A SATELLITE COMPUTER SYSTEM FOR ON-LINE ANALYSIS, CONTROL, AND DISPLAY

GEORGE H. LEACH
and
ALBERT J. PERRELLA

30

A SATELLITE COMPUTER SYSTEM FOR ON-LINE ANALYSIS, CONTROL, AND DISPLAY

****

George H. Leach

and

Albert J. Perrella

A SATELLITE COMPUTER SYSTEM
FOR ON-LINE ANALYSIS,
CONTROL, AND DISPLAY

by

George H. Leach

Captain, United States Marine Corps

and

Albert J. Perrella

Lieutenant, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
IN
ENGINEERING ELECTRONICS

United States Naval Postgraduate School
Monterey, California

1 9 6 4

A SATELLITE COMPUTER SYSTEM

FOR ON-LINE ANALYSIS,

CONTROL, AND DISPLAY

by

George H. Leach

and

Albert J. Perrella

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

IN

ENGINEERING ELECTRONICS

from the

United States Naval Postgraduate School

ABSTRACT

Existing Fortran systems for the Control Data Corporation 1604 Computer are lacking in man-machine communications capability, which limits certain desirable human program intervention and decision-making capability. Extension of the basic Fortran 60 system to provide man-machine communications and make available better problem solving techniques seemed both desirable and feasible.

This project was undertaken to achieve a problem solving complex based on the Fortran 60 system. A system utilizing a remote computing station with a data processor (CDC 160) satellited to the main computer (CDC 1604), and a cathode ray tube display unit (DD 65) used as a multipurpose input/output control unit has been designed, programmed, and demonstrated. The system features on-line, variable speed output of both printed output and graphical material presented on the display, parameter change capability in running Fortran programs, complete control of the main computer from the satellite station, simplicity in making additions and changes, and system philosophy adaptable to full scale time-sharing on the main computer. In addition, utilizing the equipment in the Electrical Engineering Department Digital Control Laboratory (A/D, D/A converters, analog computers, and plotters), this system can be extended to provide the capability of digital control of analog systems.

ACKNOWLEDGEMENTS

The authors wish to express their appreciation to Professor Mitchell L. Cotton and Professor Harold A. Titus for their aid and encouragement in the preparation of this thesis. In addition, the authors would like to thank Mr. Robert C. Brunner and Mr. Roger N. Lund of Control Data Corporation for their suggestions and cooperation concerning modifications to and maintenance of the CDC 1604 Computer.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. Introduction.

As demands upon computing facilities increase, two seemingly divergent trends are apparent. On the one hand, there is a clear need for constant improvement in the area of high efficiency, compiler-monitor operating systems featuring fast compile times, efficient object program code, concurrent input-output, and other features which are generally directed toward achieving maximum possible throughput rates. Such characteristics are best obtained in the environment of a "closed shop" service bureau type of operation. The locally available Fortran 63 Compiler and associated monitor control system is directed toward this type of service operation. Processing under this system is strictly stacked-job sequenced. Within any job, the ability to use any one of several compilers with a minimum of monitor imposed programming constraints provides great flexibility.

Despite the impressive efficiencies which may be obtained under the operating concept briefly outlined above, a significant number of computer users or potential users are concerned with applications wherein "on-line" efficiency concepts dominate any "throughput" efficiency concepts. By "on-line", we refer to situations where the emphasis is upon effective synchronization of computer operations with the action time scale of a human operator, decision maker or problem solver, or alternately with the time-scale and dynamics of a natural or experimental process. Without attempting to delineate the bounds of justifiable "on-line" use of

expensive computing machinery, we note only that the strongest and most basic justification lies in the viewpoint that the most valuable roll of the computer is that of an extender and augmenter of the problem-solving ability of the human user. In a Navy in which the general purpose computer is rapidly becoming an everyday tool of command and control, it is not surprising to see an increasingly high level of interest in "on-line" computer usage.

While there seems little doubt that the best features of the two trends described will eventually be combined through advances in operating system concepts, engineering design, and language improvements, a considerable amount of work remains to be done to make the promise a reality.

The objective of the thesis project described in this paper has been to undertake, within the constraints of currently available equipment, the implementation of a system having several useful on-line features. Also, certain basic elements of time and space sharing are provided with provision for future elaboration. The system to be described has the following features:

1. The operating system includes a basic compiler and compatible symbolic assembly system.
2. The system characteristics are provided by additions to rather than modifications of an existing control system (Fortran 60).
3. Provisions have been made for Task Oriented Query and Response service to either of two satellite

computer stations under the censoring action of an executive routine.

4. Provisions have been made for centralized library file storage and call service for the satellite computer stations.

5. Effective main computer control and highspeed direct computer output (both graphical and text) via a CR-tube display console with associated function and typewriter keyboards has been incorporated.

6. A flexible system of updating and maintaining magnetic tape files (typically private program files) and an on-line tape editing capability have been provided.

7. A capability of on-line parameter change in running programs has been developed as a problem-solving aid.

8. Provision has been made for incorporating a large computer on a time-shared basis into bench experimental setups via an Analog-Digital conversion system.

2. <u>The Fortran 60 Compiler and Control System</u>.

The Fortran 60 Monitor System is a simple and efficient system for processing batched jobs on the 1604 computer in the Control Data Corporation 1960 version of the problem oriented Fortran language. This basic control system with its associated compiler has been the mainstay of U.S. Naval Postgraduate School computer operations for several years. That the system accomplishes its task of providing adequate service for the majority of users is not in

issue, but simplicity and relative efficiency should not be permitted to mask the fact that certain deficiencies do exist. A discussion of the advantages and disadvantages of the system from a systems operating viewpoint may be found in (16).

Under any programming system oriented primarily toward throughput efficiency, all program decisions must be made prior to compilation by the programmer-problem solver. While the program is running in currently available systems of this type, no communication between the operator and the program is available. Programs must run to completion, and output must generally be inspected off-line. Any rudimentary form of parameter optimization requires that many sets of runs be made, changing parameters by programming. In many cases, a human decision made at the proper time during the course of a running program could not only provide the path to a quicker solution, but could also decrease the actual amount of computer time used by a more conventional approach.

Realizing this seemingly simple goal requires not only the ability to have all computer output presented visually to the operator-problem solver, but that means of complete control of computer functioning be available in conjunction with these visual results.

Newer versions of the Fortran language have made their appearance along with more complicated and supposedly more sophisticated control systems. None provide the ease and simplicity for system changes or additions better than the Fortran 60

control system. That the associated compiler of this system lacks some of the language sophistication of later versions can be accepted when we consider that this system provides an essentially one-to-one symbolic machine language for systems programming and equipment manipulation. Since this system is sufficiently valuable, it has been used as a base for logical extension to provide additional on-line services, such as high-speed visual display and graphing, parameter change capability, and operator control and communications through the display device. The control system has been extended with complete compatibility with the original system,and operator-programmers may make use of the additional features with few operational or programming constraints.

As an introduction to the system philosophy, and to permit intelligent utilization of the system documentation provided in the accompanying appendices, a brief discussion of the Fortran 60 Resident Control System will now be presented.

The Fortran 60 resident control routine is composed of a series of closed subroutines, which provide the basic control of the computer and of the assigned external equipment. When loaded and running, the program senses the console typewriter for instructions. When completion of an instruction statement is signalled by typing a period, the statement is processed and the computer responds by performing the indicated operations or by delivering some communication to the operator via the typewriter. When the operation is completed, the computer returns to the typewriter wait loop.

Processing the usual program which is involved in solving a specific problem is most easily accomplished under the Monitor control routine. In this case as before, the computer responds to the control statement by loading the compiler above the resident bias level. Control is then passed to the compiler, and the job is compiled in memory. Certain closed subroutines of the resident program are available to the compiler, as they are to any program, through certain fixed low core entry/exit cells. When the compiler desires a certain equipment function, it merely performs a return jump to the proper low core entry cell. When compilation is completed, the job is assembled in core above the compiler, and control is returned to the resident routine. Control is immediately passed to the job itself where it remains until the job is completed, at which time it returns to the resident control routine. (See Appendix I-A for jump cell particulars)

FLOW CHART SYMBOLOGY

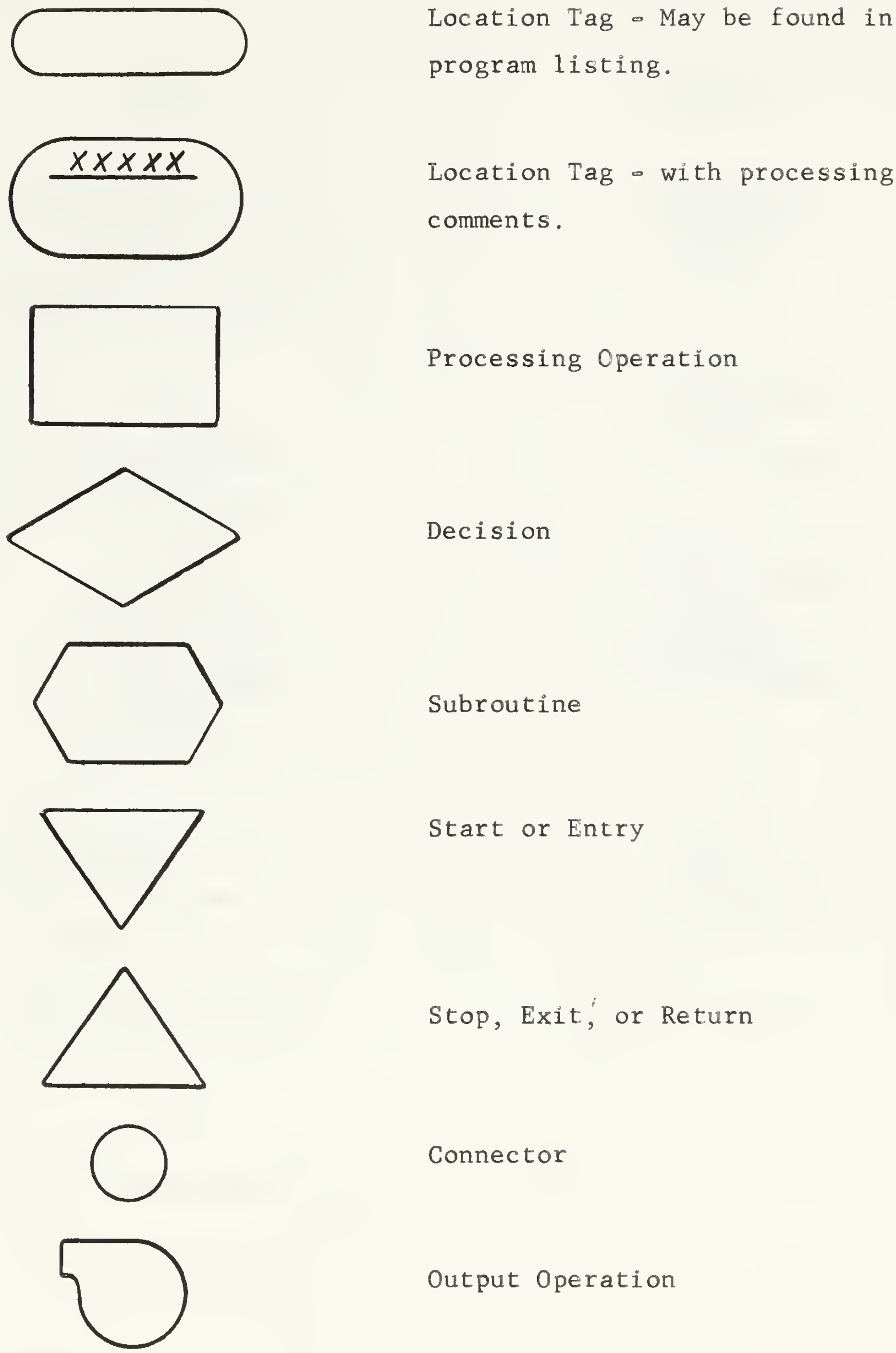


FIGURE 1

SIMPLIFIED FORTRAN 60 CONTROL AND INTERRUPT PROCESSING

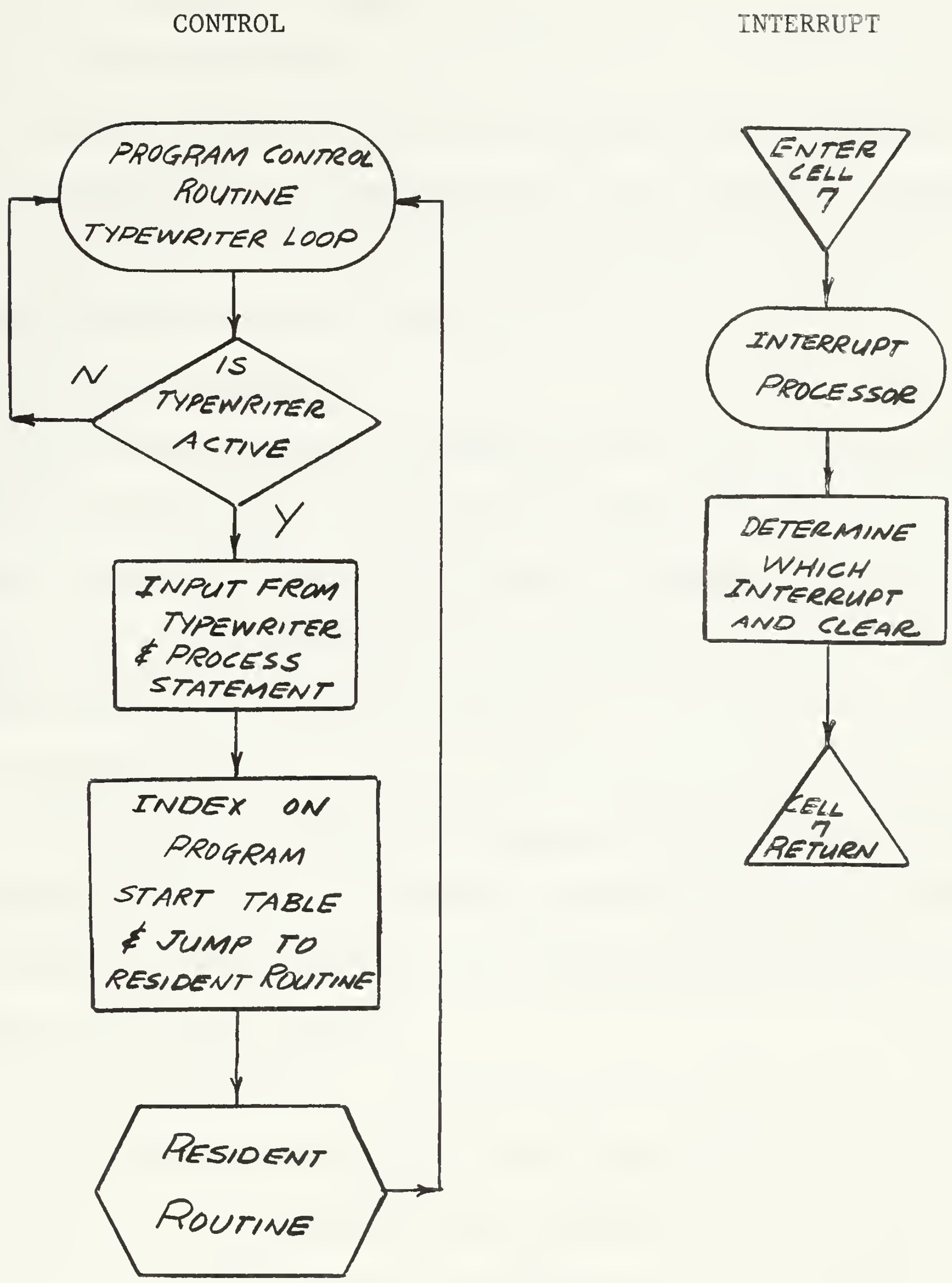


FIGURE 2

Acquiring a thorough knowledge of the entire resident system is time consuming. The routine is divided into areas by location tag and may best be described by a few words on each of these sections which are immediately pertinent:

OZ - Program Start Table.

This table consists of a list of addresses of the resident routines which may be entered directly from a typewriter statement. The table also contains space for the addresses of transient programs loaded by the Call routine.

1A - Time Advance Routine.

This section provides for advancing the computer clock once each second when the clock cell at location 00000 overflows. The clock cell is incremented every 1/60 second by independent circuits. This routine also determines if a time limit is exceeded. For hardware details, see Page 5-41 of (2).

3A - BCD Read Switch.

This segment performs the linking operations to the BCD read routines. If the Monitor Routine arguments are set, all reading is done by coding imbedded in this routine without benefit of parity or length checking.

4A - BCD Write Switch.

This segment performs the linking operations to the BCD write routines. If the Monitor Routine arguments are set, all writing is done by coding imbedded in this routine without benefit of parity or length checking.

5A - Binary Read Switch.

This segment provides linkage with the binary reading routine.

6A - Binary Write Switch.

This segment is similar to the binary read switch.

7A - Backspace Switch.

8A - Rewind Switch.

9A - Endfile Switch.

These switch routines provide links with the appropriate equipment handling routines.

11A - BCD Message Switch.

This segment provides output processing of packed BCD messages. Typewriter is used if no Monitor Routine arguments are set; otherwise the output medium is used.

1C - Call Routine.

This routine performs a search of a binary format tape for designated service routines or programs, and loads these into core when found. The load location is entered in the Program Start Table (OZ) so that they are then available for immediate execution by a typewriter statement.

1E - Error Message Handler.

This segment channels messages concerning equipment failure.

3E - Output Packed BCD Message.

This coding links with error indicators 1E and 2E and performs the function of outputting a packed BCD message to the proper equipment.

1G - Input Index Register Loader.

This routine sets index registers 1 and 2 to values based on the quantity in the A register when entered. These index registers then determine the proper path of execution through the various input (read) routines. They are also used to build the necessary external function codes for the equipment specified.

2G - Output Index Register Loader.

This section performs functions similar to 1G for output (write).

3G - Disassemble Read Buffer.

This coding segment disassembles the read buffer from a packed group of 15 cells (120 characters) to a buffer of 120 cells, one character per cell.

4G - Assemble Write Buffer.

This coding segment assembles 15 packed cells from 120 cells containing one character each.

5G - Sense Typewriter for Action.

This routine outputs a message in typewriter code and then sets up a loop waiting for an operator response. The response is then processed and the A register loaded with some code, which is carried back to the routine which initiated the return jump to 5G. This value is then used to determine a course of action.

6G - BCD Typewriter Output.

This routine performs the function of outputting a packed BCD message to the typewriter.

7G - Typewriter Print Routine.

This routine actually performs the equipment handling for output to the typewriter.

8G - Assign Tape Medium and/or Sense All Channels Inactive.

This routine determines unused but ready tape units for scratch writing and reading under Fortran statement "Write output tape n". When entered with zero argument in A register, the routine senses all buffer channels inactive.

1I - Read Binary Record.

1O - Write Binary Record.

These are routines which perform the packing and unpacking of the standard 54 word blocks of the binary tape format.

1M - Monitor Routine.

This control routine is used to process batched jobs.

1P - Program Control Routine.

This routine provides the basic operator-computer communications link through the typewriter, decodes control statements, packs arguments for resident routines or routines listed in the Program Start Table; then passes control to the program designated in the statement.

2Q - Hold Routine.

This section provides for flag setting to signal the Monitor routine to backspace the input tape. Also increments the initial number of programs and the initial bias to the current values, as found in the control information table found at absolute location 00060.

3Q - Clear Routine.

This routine clears all memory above the bias level to memory location 77777.

1R - BCD Read Routine.

1W - BCD Write Routine.

These routines handle the packing and unpacking for the read and write routines respectively. Card reader/punch and paper tape reader/punch equipment handling is imbedded in these routines. Tape handling is accomplished in secondary routines entered through these routines.

1X - Tape Reading Routine.

1Y - Tape Writing Routine.

These routines perform the actual equipment handling of the tape units and are entered from the binary and BCD read/write routines. Full provision is made for parity and length checking, as well as sensing endfiles and end of tape. These operations are unbuffered.

1V - Rewind Specified Medium.

3V - Rewind Tape With Interlock.

5V - Write Endfile on Specified Medium.

7V - Backspace Specified Medium.

These routines perform the actual equipment handling to accomplish the specified task. They are entered from the corresponding switches previously discussed.

run, and results returned. The fact that the two were separated by some 120 feet of cable did not alter the programming requirements, nor the validity of any conclusions as to logical extension of the system.

The arrival of the Data Display DD-65 Multi-purpose Display Unit in early 1963 provided some additional equipment capability not previously available. Examples of the use of this display unit may be found in (16) and (17). The first of these references shows a use of the display in conjunction with normal tape output as a simulated line printer. The application was attractive but rather inflexible. The condition of the unit at this time did not allow for completely trouble-free operation, and many cases of main computer hangup, due to faulty operation of the display unit, provided conflicts with Computer Facility operations.

The second reference was a simple extension of (15) to demonstrate the capability of the display unit to function as both an input and output device for remote Fortran programming, although extremely inefficient for input and lacking hard copy capability for output.

The uses demonstrated by (16) properly developed the satellite capability of the CDC 1604 - 160 computers and did much to uncover certain deficiencies in the inter-computer communications logic not covered in the CDC programming manuals. The work accomplished in (16) showed what could be done, but unfortunately as demonstration tools, they lacked a coherent base for their

overall philosophy. Changes or additions to the system, as accomplished in (17), required a complete knowledge of the resident control system and all system modifications, and also required inordinate amounts of time.

What was needed was a system that had been logically developed on a well founded philosophy, where proper and complete documentation would provide users and systems programmers with easily acquired tools for additions and changes.

4. Factors Influencing System Advancement.

Several subject areas influence any attack made on the problem of extension or advancement of the Fortran 60 system. Each is concerned with both hardware and philosophy. The subject areas may be defined as follows:

4.1 The capability of the control system and its interrupt processor.

4.2 The satellite processing station concept.

4.3 The capability of available equipment.

4.4 Multi-satellite operations and time-sharing.

It was the authors' desire to impart sufficient modularity and flexibility to the system structure that the addition of new features or changes to the system might be made with no fear of disturbing other processes. Each of the subject areas mentioned above contributed constraints and direction to the development of an improved system philosophy.

4.1 Capability of the Control System and its Interrupt Processor.

The interrupt processor under Fortran 60 possessed only the capability of recognizing arithmetic interrupts, clearing these interrupts and returning to the main program. The sole functional use of this interrupt processor was for incrementing the computer clock each second. See (2). The work accomplished in (16) extended the interrupt processor to recognition of satellite computer interrupts. The access scheme used in this work required that the main computer be processing under Monitor control, and that between jobs of a batched job stack, the satellite would be allowed to enter for processing of its Fortran program. This required a wait for service until the job in process was completed. Since this scheme possessed only one mode of operation (short Fortran programs entered via paper tape input medium at the satellite computer), the wait was of little consequence. The difficulty here lay in the fact that the main computer was not accessible unless it had been started at the main console. No attempt was made to achieve active control of the main computer from the satellite station. During the authors initial studies of Fortran 60 resident control system, it became obvious that a more sophisticated interrupt processor would be needed to achieve any improvement in control. The writings of Codd (18), and the current high interest in the data processing industry in multiprocessing and multiprogramming, suggested control philosophies where more than one processor is available. These philosophies are based on an

executive control routine, which makes decisions as to which operation or job will be next processed.

4.2 The Satellite Processing Station Concept.

The multiprocessing aspects of utilizing more than one computer points to separate computing stations with limited capability at the satellite and maximum capability available to a variety of users from the main computer. The satellite unit has uses that permit it to operate independently of the main computer, but certain operations deem it extremely attractive to have the services of the main computer available at the satellite station. In this pseudo time-sharing context, it seems advisable that control of the main computer be exercised from the satellite station when used in performing functions specifically designed for primary main computer service to the remote station. Any implementation of a true full scale time-sharing system would,of course, control processing with the main processor executive routine.

Since the satellite processor has capability of its own, it seems attractive to have its library of routines stored on the library of the main processor, so that they could be called into memory by a short bootstrapping arrangement, obviating the use of paper tape loading of the 160 computer.

4.3 Capability of Satellite Station Equipment Available.

The Digital Control Laboratory of the Electrical Engineering Department, U. S. Naval Postgraduate School, maintains a CDC 160 Computer, a DD-65 Display Unit, several analog computers,

A/D and D/A converters, and an X-Y analog plotter. This equipment is separated from the CDC 1604 Computer located in the Computer Facility by some 120 feet of cable, but substantially trouble-free data transfer over the cables had been achieved in (16), where 160 and 1604 computers communicate via the 1607 tape units (13).

The capabilities of the CDC 160 Computer are described at length in (8, 9, 10, 11).

The Data Display DD-65 Display Unit is a general purpose double cathode ray tube arrangement which accepts inputs from an associated computer and/or radar set. The memory of the display is non-accessible to other equipment and must be packed by output from the associated digital computer. All display formatting must be accomplished by logical processing in the associated computer before updating the memory of the display, where the character and vector generating circuitry performs the necessary beam deflection and blanking to produce the cathode ray tube presentation.

The 1604 computer has the capability of transmitting and receiving high speed digital data on transfer channel #7. This is a non-buffered data channel not capable of simultaneous input and output. The DD-65 Display Unit is configured to utilize this direct mode of data transfer with the 1604, as well as operation with the 160. For remote display of information and rudimentary control of the 1604, this direct linkage would be adequate.

However, in order to provide a multiprocessing capability and adaptability to full time sharing operations, the use of the 160 as both an equipment controller and communications buffer between the display and the main computer seems far more advantageous. This usage allows less complexity and more flexibility in the main computer control system, removes the need for continuously monitoring the display for input, and would seem to greatly reduce the amount of resident programming required for display processing. The burden of specialized handling of information may then be placed on the 160, since it has access to all additional equipment which may be found in the remote station setup. This requires an executive routine for the 160, means slower data transfer, and adds complexity to program timing between the 1604 and the 160, but permits a main computer control system which is independent of any equipment located at a satellite station except the 160.

The analog computers A/D and D/A converters, brush recorders, and X-Y plotter could provide a great potential in control system problem-solving and experimentation if linked to a central high speed processing system to which the services of the 1604 computer are available.

4.4 Multi-Satellite Operations.

It would seem that Fortran 60 improvement should allow for more than a single satellite processing station to permit other departments to acquire service for their specialized requirements

in a manner discussed in the previous section. The 1604-160 communications organization immediately allows for two stations - one each on channels 3-4 and channels 5-6. See (13). Under a fully implemented time-sharing system, this number could be reasonably increased to six to eight by using code words to determine toward which satellite station a communication is directed.

The executive control routine of a system with some of the characteristics and capability discussed previously need be little more than a traffic controller; but a system thus founded has the advantage of simple extension without changes to the other associated functions. The main computer control philosophy should be general. Specialized equipment at remote stations could then be integrated with the satellite data processor control system, with no changes necessary to the main computer resident control system.

The previous discussion points toward a task organized philosophy under the direction of an executive controller which will determine which task and when, and individual closed subroutines to perform the desired tasks. These tasks will provide services for a satellite processing station with operational capability of its own, and provide the necessary linkage for direct control of the main computer from the satellite station.

5. System Implementation.

5.1 Main Computer.

The desire to maintain complete compatability between a new system library and the library currently used by the Computer Facility was based on the authors interest in promoting the use of their system as the standard Fortran library for use in the Computer Facility. In all cases of programming and operator usage, no difference between the new and the old systems is apparent. Details of the modifications to the basic Fortran 60 resident may be found in Appendix I. The modified control system has been named FORTSHARE.

5.1.1 Achieving Main Computer Service for the Satellite Station.

A timed typewriter wait loop was established whereby the typewriter is sensed for action for a short period. If the typewriter is not busy within this period, an overflow interrupt is generated to secure a path through the executive routine to inspect for satellite requests for service which have been set by the interrupt processor. See Figure I-1.

Interrupt processor operations have been extended, not only to recognition of the satellite system interrupt codes but to logging which satellite station interrupted. The logging of a satellite interrupt in either of two flags is then used by the executive routine to determine if satellite service has been requested. See Figure I-2.

The interrupt processor always exits to the executive routine whose function it is to determine if satellite service has been requested, and if so, whether or not it can be granted at that instant. This is accomplished by checking values of the three control flags used in various places throughout the resident program to suppress satellite service during certain critical operations. No service is permitted during any input-output operation (AOK flag), during the short portion of the Monitor routine where a new job is being set up (MON flag), and during any satellite read-write operations (FLAG flag). These constraints prevent immediate service to a satellite request in some cases, but sensitivity is at most a matter of milliseconds under normal operations. Two periods are critical and require a longer period of lockout. During the Call routine, service to the satellite stations could possibly occur during the time that the routine was actually occupied in loading memory. Here satellite service has been locked out after the library search has found its first called program. The compiler uses its own call routine, and since no access to this routine is available from resident, all satellite service is discontinued during compiler search for subroutines. These constraints have proved to be of no obstruction.

An additional flag has been used for allowing the executive routine to make an immediate decision as to service when the interrupt processor is entered via the timed typewriter wait loop.

SIMPLIFIED 1604 SATELLITE REQUEST PROCESSING

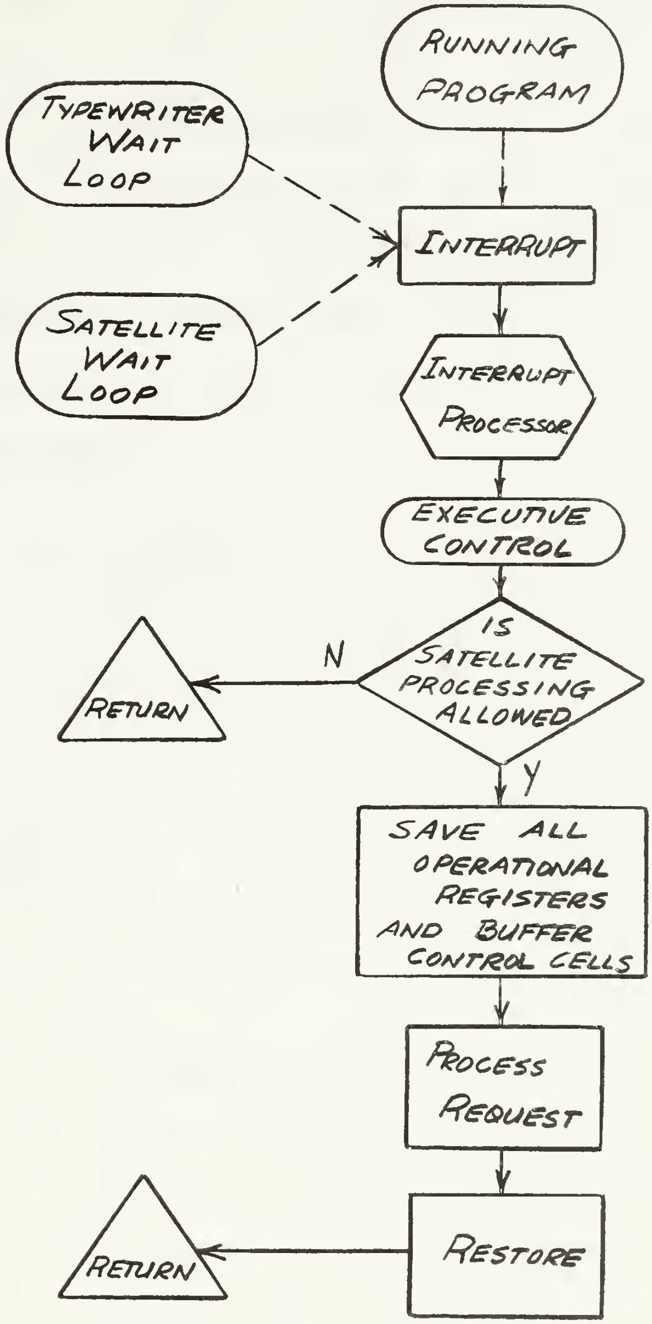


FIGURE 3

Here the requested service may be granted immediately, since the main computer had been at idle to permit this entry. See Figure I-3. Setting Jump Key 1 at the 1604 console prevents 1604 response to all satellite requests, effectively locking out all satellite operations.

All input-output operations are routed through a standard exit processor which generates an overflow interrupt, as in the typewriter wait loop, to provide a path through the executive routine to permit inspection for satellite requests. This gives good sensitivity to requests during long, continuous output at the main computer.

5.1.2 Task Organization.

When service is permitted, the main computer interrogates the satellite station for one 48 bit word, which determines the task desired. When a request is honored, all operational registers and buffer control cells are stored. When the task is completed, all are restored before exiting the interrupt processor routine. See Figure I-5. All tasks presently used in the system are described in Appendix I.

5.1.3 Satellite Communication Packages.

The CDC 1604-160 satellite logic provides an immediate capability for two satellite stations, as previously related. System programming in all instances reflects dual capability for either station. Previously installed cabling from the Digital Control

Laboratory to the main computer site is associated with channels 5-6, while the 160 computer adjacent to the main computer in the Computer Facility is cabled to channels 3-4 and may be temporarily connected to channels 5-6 if desired. In a few instances, programming may be thought to be tailored to the requirements of channels 5-6 because of the equipment available in the Digital Control Laboratory, but it is, in all cases, completely general in format, and may be processed by any Satellite Station based on equipment available at the station.

The read-write selections for direct transfer of data over the buffer channel pairs have been included in the resident programming as closed subroutines. See Figure I-6. For access by programmers, these routines are available through a standard low core entry cell, with arguments carried in the A and Q registers on entry. See Appendix I.

Normal read-write operations under Monitor control are conducted completely buffered. This feature is used only in this location in the Fortran 60 system. All read-write operations under Monitor take place without parity or length checking, which allows the buffered operations. When implementing linkages to the satellite read-write routines, it became necessary to wait for all activity on associated channel pairs to cease because of the interaction between read-write pairs during direct transfer. See (13). When satellite operations are being conducted, the buffered operations of the Monitor routine are effectively unbuffered, slowing operations some 10 to 20 per cent, depending on the quan-

tity of output. This seems to be a fair price for maintaining fully buffered operations when using the system for processing batched jobs.

5.1.4 Achieving Full Control of the Main Computer From the Satellite Stations.

The capability for operations and control are completely equivalent at either of the Satellite Stations, but the availability of the DD-65 Display Unit at Satellite Station 2 has influenced the mode of operations at that station. It must be restated, however, that all control means from and to the main computer are general. Any variety of equipment at a Satellite Station may be included in that system, with no change whatever in the structure of the 1604 resident.

The problem of satellite communications had been solved with the addition of the packages mentioned in the previous subsection. The use of the typewriter as a control medium has been paralleled in satellite control. All entry/exits for control communications to or from the operator via the console typewriter are diverted to the Satellite Station in control. For output, these include BCD packed messages, typewriter code packed messages,followed by sensing the typewriter for operator action, and normal typewriter output via the BCD write routine. For input, these include normal input, with the typewriter acting as an input medium, entry of a completed typewriter control statement into the Program Control routine, and input generated by

operator action when sensing the typewriter for action.

A control flag (SAT flag) has been used to perform this diversion in each of these input-output locations. This flag is multi-valued to cause jumps to the necessary routines for packing and communications with the satellite station. (SAT = 0 for main computer control, 1 for Satellite Station 2 control, and -1 for Satellite Station 1 control). These routines make up the major additions to the Fortran 60 resident control system. When satellite communications are completed, a return is made to the proper location for continuing the program. In effect, the main computer has had its control medium changed, but is unable to determine any difference.

Setting the SAT flag by the satellite processor diverts the normal typewriter wait loop under Program Control to another wait loop, which will be referred to as the satellite wait loop (SWL). When in the SWL, the computer is at idle under control of one of the satellite stations, in much the same manner of the TWL. This loop periodically jumps out to generate an overflow interrupt to gain a path through the executive routine to inspect for satellite requests. Imbedded in this wait loop is an action flag, which provides a signal for the main processor to jump out of the SWL and input a BCD card image from the satellite station. See Figures I-7 and I-8.

5.1.5 Computer Output Processing.

Normal program output generated by the main processor is

delivered via the BCD write routine. Included here is a programmed divert switch to send 64 characters of the 120 character write buffer to the satellite station for processing to its display. Control of the main computer is not necessary to use this feature; thus, both satellite stations may view main processor output simultaneously. See Figure I-9. This feature slows operation of the main processor slightly by effectively unbuffering Monitor read-write operations, but provides direct operator viewing of main computer output without the necessity for off-line processing. Output is displayed in a moving page format discussed in Section 5.2.1. Selectable operation of this feature is important when considering the concept of system usage and operator decision-making capability.

5.1.6 Additional Routines.

In order to effectively utilize the tools offered by the control system programming, as implemented in the FORTSHARE resident control routine, several normally callable routines have been added to the Fortran 60 library repertoire.

5.1.6.1 CHANGE.

This program callable subroutine enables a user to change specified variables in a running Fortran program on the console control medium, whether at the main console or a Satellite Station. This routine has great utility when attempting to converge on an optimum value of a parameter, or in program branching, depending on the judgment of the operator. It also enables a user to eliminate data cards from an often used program, since

values may be typed in directly. See Appendix IV.

5.1.6.2 SATGRAF.

This subroutine provides for on-line graph display at Satellite Station 2, utilizing the DD-65 Display Unit. Graphical display is made on the left-hand tube of the display unit simultaneously with BCD information on the right-hand tube. Use of this routine in conjunction with CHANGE makes it possible to immediately view the effects of parameter changes. See Appendix V.

5.1.6.3 File Maintenance.

Several routines to aid users with file maintenance problems have been added. Routines entitled FILEIN and FILEOUT provide a means of building BCD files and quickly securing the information contained in a specific portion of the file. See Appendix VIII for detailed treatment of these routines.

The most important feature of file maintenance capability provided with FORTSHARE is the addition of service routine EDIT. This routine is typewriter controllable from the Satellite Station possessing the DD-65 Display Unit. The routine provides the capability of on-line editing of BCD tapes. Thus, program input tapes may be corrected on the computer without the necessity for off-line processing of an output tape to find errors and retaping a new input tape after making corrections to program cards. See Appendix III.

5.1.6.4 ANALOG and PLOTTER.

These routines provide communications with the analog

devices in the Digital Control Laboratory. ANALOG provides program selectable sampling of the analog to digital converter or transmission of data to either of two digital to analog channels. PLOTTER will plot an on-line graph on the X-Y plotter through the digital to analog converters.

5.1.7 Remote Console Wiring.

Due to the physical separation of the Computer Facility and the Digital Control Laboratory, where the fully implemented Satellite Station 2 has been installed, it was considered desirable to have some form of manual control over the main computer when the control of the main computer resided at Satellite Station 2. The possibility of malfunction at either the main or satellite processor, destroying the timing of the integrated control programs, could not be discounted, and the ability to clear and restart the main computer from the remote station would solve the problem.

In order to implement this control, a 26 wire cable was laid parallel to the present channel 5-6 cables from the remote station to the 1604 console, approximately 130 feet. Twelve wires of the cable were used to parallel the 1604 console switches for START/STEP and MASTER CLEAR/EXTERNAL CLEAR and the 10 and 20 bit punches of the program address register. In addition, control lights indicating whether the console lights are on or out (computer running or stopped), channel 3-5 active (read), and channel 4-6 active (write) lights, have been wired. A provision for wiring in an autobootstrap feature has been left available by utilizing the

present wiring for the 20 bit punch of the program address register. This control from the remote station allows complete console operation of the 1604 Computer, with the exception of visual contact with register values and the tape units. See CDC Maintenance Modification Number 1604 USNPGS - 002.

A further aid to the satellite operator is the use of the two console lights on the Digital Control Laboratory 160 Computer, which had been previously installed (16). The logic controlling these features was added to the 1607 unit of channels 5-6 to permit program selection of the Program Control mode of operation without physically switching the COMPUTER SELECT SWITCH out of the 1604 ONLY mode. See (13). The logic also provides for operation of the two console lights on the satellite 160 Computer located in the Digital Control Laboratory.

The acquisition of new 1607 Tape Units during March and April, 1964, required reinstallation of this modification. This has been accomplished in compact form for channels 5-6, and an identical modification to channels 3-4 may be made when deemed necessary. Full documentation of this modification may be found in CDC Maintenance Modification Number 1607 USNPGS - 001. Programming applications may be seen in (19).

5.2 Satellite Data Processor.

The Digital Control Laboratory of the Electrical Engineering Department, USNPGS, provided the complex of equipment about which the authors developed the philosophy of remote station control.

The availability of equipment here, specifically the DD-65 Display Unit, influenced the implementation of the remote station control means and usage. It should, however, be restressed that all means of data delivery to satellite stations, and all format of data delivered is completely general and in no way restricts the equipment which can be used.

5.2.1 The Executive Routine for the Satellite Data Processor.

The satellite data processor (CDC 160 Computer) acts as the remote station controller. All input and output to and from the satellite station must be secured and directed by this equipment. The display associated with the satellite data processor has all its inputs and outputs secured and processed by the 160.

The philosophy of the executive routine is basically that of a multistation sensor. The routine alternately samples the main computer and the display unit for input. When an input is present for the satellite processor, the input is accomplished and disposed of according to the structure of the routine. Normally, input from the display is control input from Keyboard 2, or a character from Keyboard 1, which must be processed for display. Input from the main computer is always a 64 character buffer, or 48 bits of prepacked vectors, depending on the condition of communications flag one in the main computer. A 64 character buffer is always BCD information, which is displayed as one line. See Figure 4.

The display console is the control medium for the entire satellite station complex. Keyboard 2 consists of 30 keys with

associated logical signals, while Keyboard 1 consists of an extended alphanumeric typewriter style keyboard with standard BCD character representation. Subroutinized tasks have been assigned the keys of Keyboard 2. Format chosen for display is a 16 line rolling page similar to that of an electric typewriter or a high speed line printer. The new line is printed at the bottom of the page, while all old lines are incremented up one line. In the case of the display, the top line is merely discarded, that is, it appears to roll out of sight at the top of the display screen. For type composition, the carriage return increments all lines up the screen. When a line of data is received from the main processor, the screen is automatically incremented, so that the last line received is the most current.

5.2.2 Keyboard 2 or Function Processor.

The second major division of the control routine for the satellite complex concerns Keyboard 2 and specific program functions. Description of the functions performed here are covered completely in Appendix II. Layout of Keyboard 2 is illustrated in Figure II-1.

5.2.3 Main Computer Control from the Satellite Station.

Control of the main computer from the satellite station is identical in all respects to control at the main console, with the exception that complete control statements are typed to the display screen before they are entered into the main computer. See Appendix II. Operator communications are accepted by the satellite system in lieu of typewriter output when the Satellite Station

is in control. These statements are displayed as the bottom line of the rolling page. Aids to control have been implemented with the main computer tape status routine (See Appendix I), and the simplified remote console installed at Satellite Station 2.

When the output of the main computer is viewed at the satellite station display unit, only 64 characters are seen. This constraint is imposed because of display readability considerations, but is of little consequence, since problem solvers using the system may easily limit output to 64 characters and still have ample room for their needs.

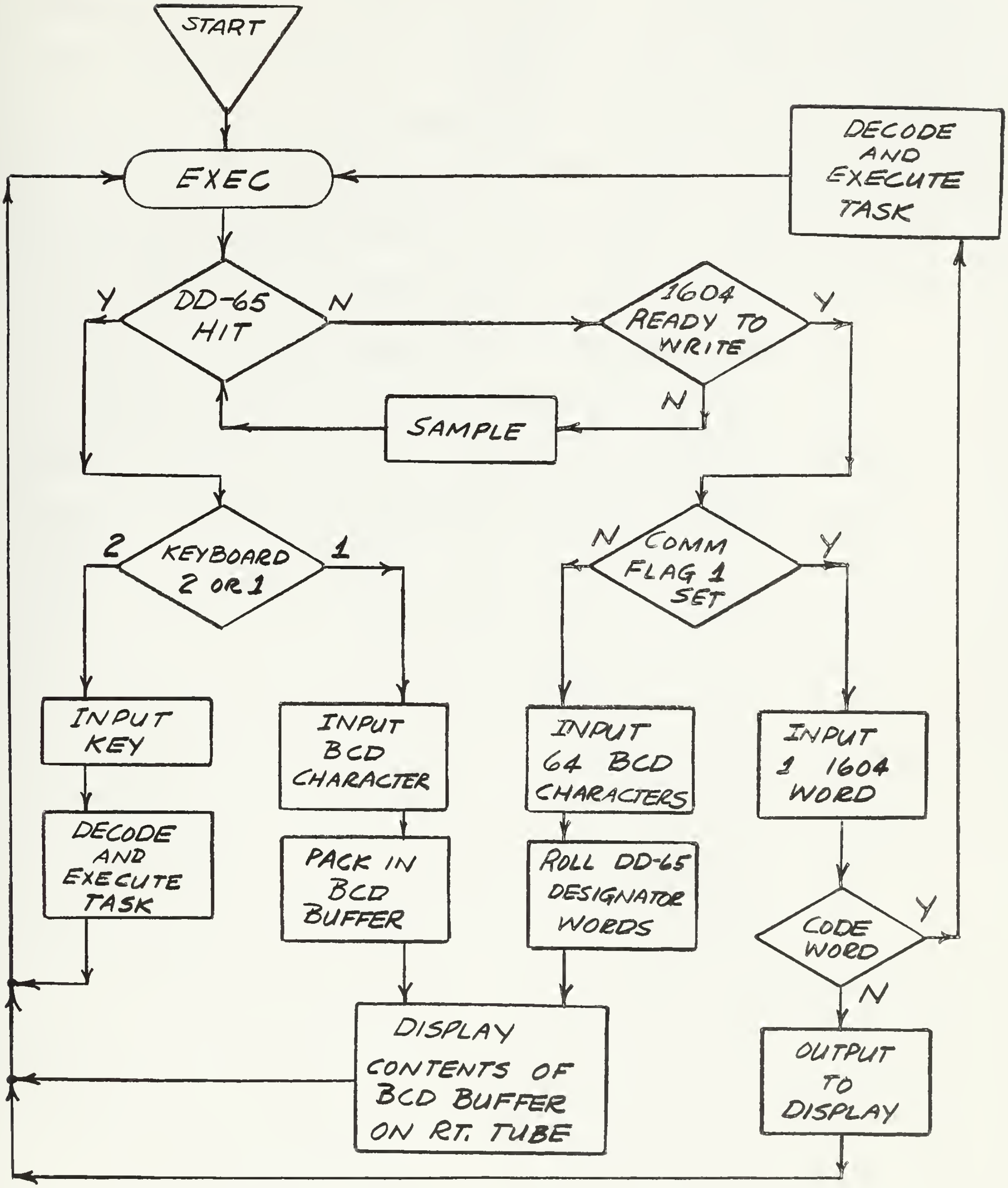

START
EXEC
DECODE AND EXECUTE TASK
DD-65 HIT
Y
N
1604 READY TO WRITE
Y
N
SAMPLE
KEYBOARD 2 OR 1
2
1
COMM FLAG 1 SET
N
Y
INPUT KEY
INPUT BCD CHARACTER
INPUT 64 BCD CHARACTERS
INPUT 1 1604 WORD
DECODE AND EXECUTE TASK
PACK IN BCD BUFFER
ROLL DD-65 DESIGNATOR WORDS
CODE WORD
Y
N
DISPLAY CONTENTS OF BCD BUFFER ON RT. TUBE
OUTPUT TO DISPLAY

FIGURE 4

5.2.4 Extension of the Basic Executive Control Routine.

The original concept for the satellite station executive control routine was to make it sufficiently general to allow additional equipment functions or program functions, with no changes to the philosophy or programming of the basic control loop. This concept has held during the months of the system development. The addition of special system features, such as graph plotting, use of the A/D and D/A converters, and the use of flex tape as a Fortran input, have required no changes, merely additions.

Appendix XI deals with the mechanics of maintenance and change to system control philosophy and programming for both the satellite station data processor and the main computer control routine.

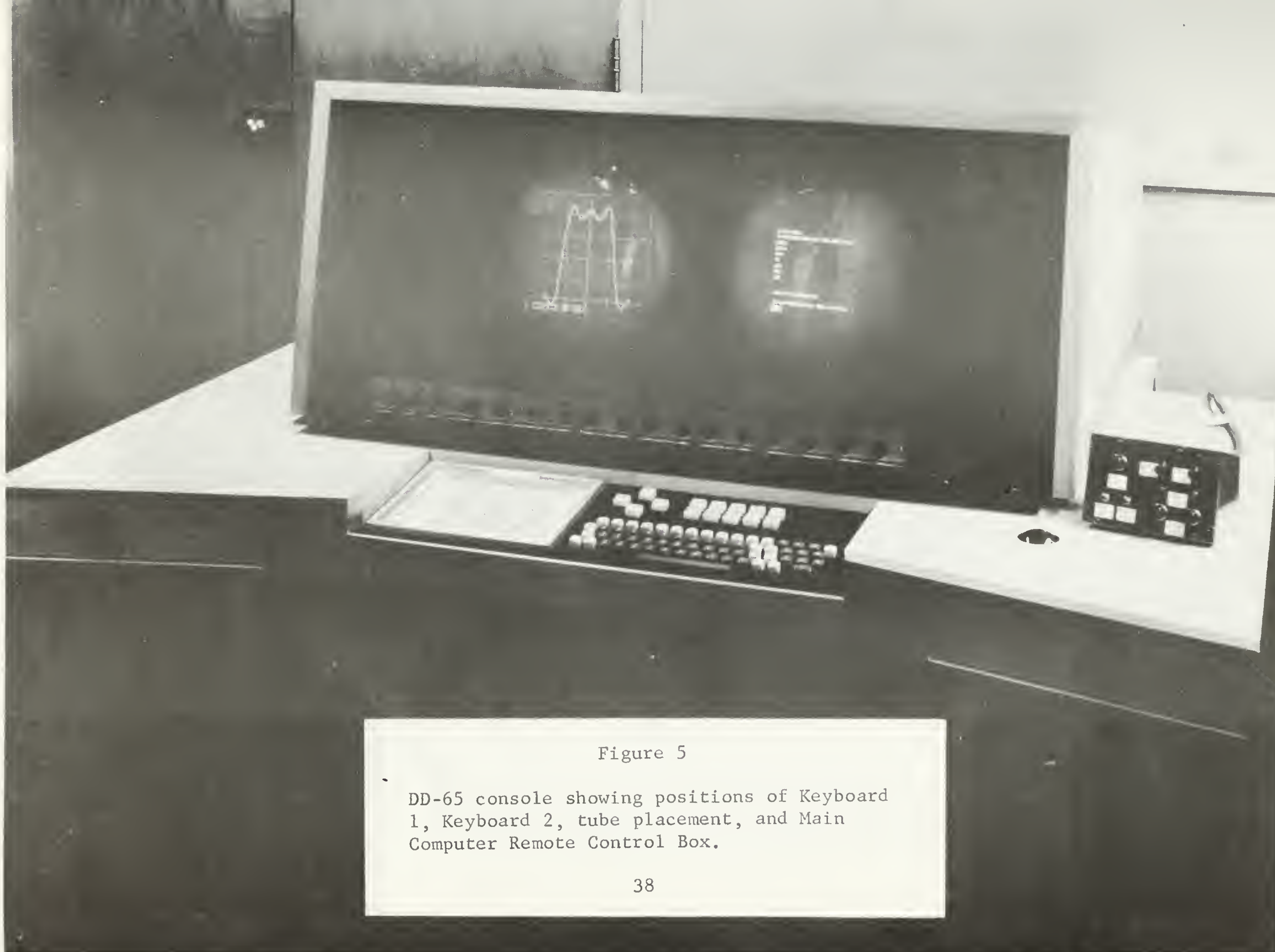


Figure 5

DD-65 console showing positions of Keyboard 1, Keyboard 2, tube placement, and Main Computer Remote Control Box.

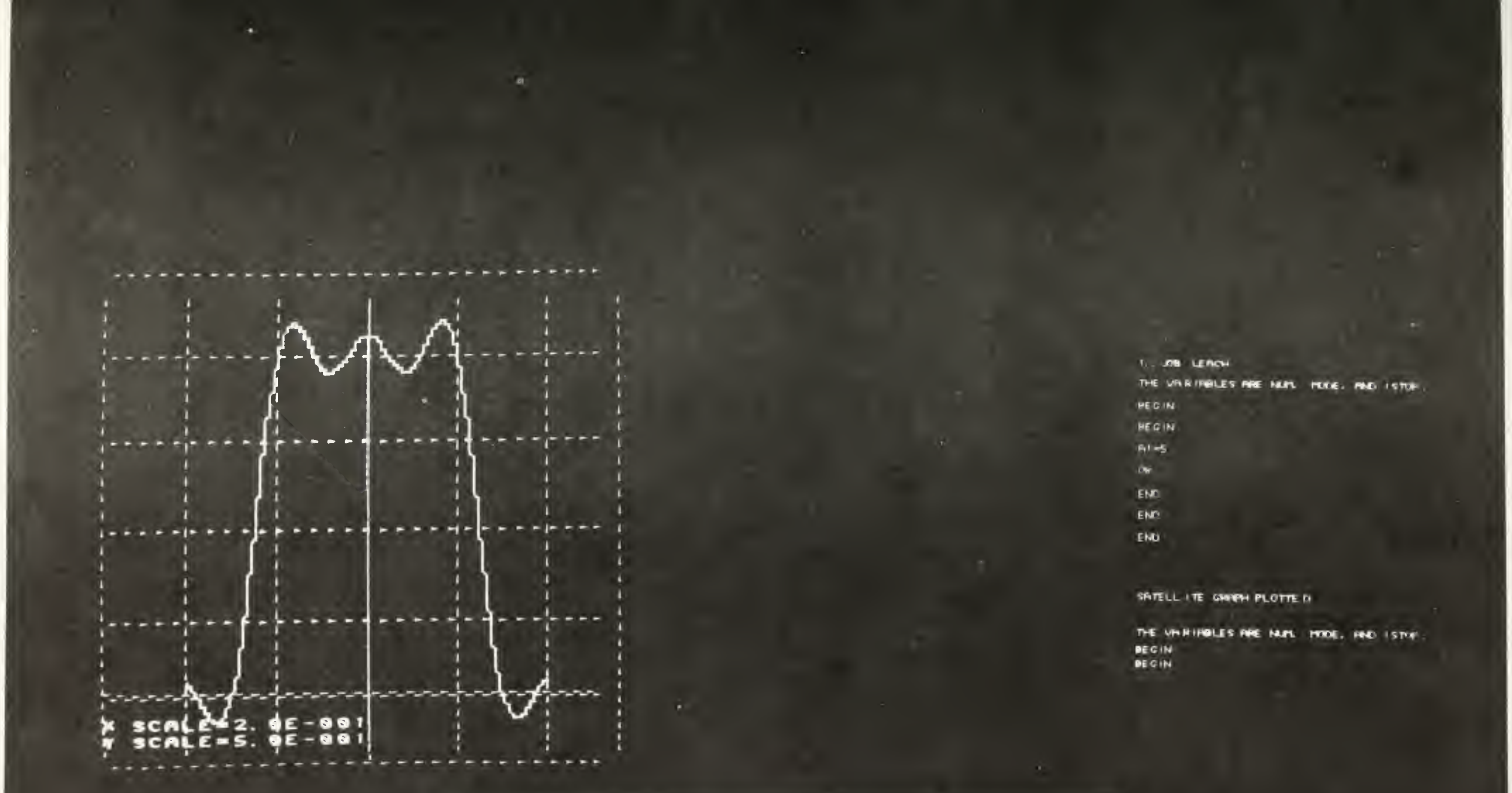


Figure 6

BCD control and output presentation (right tube). Graphical display (left tube). Note graph roughness due to vector direction and length limitations of the DD-65.

6. Equipment and Logical Difficulties.

Some difficulties were encountered in system implementation which could not be foreseen. These were the results of both equipment logical shortcomings and equipment interaction.

6.1 CDC 1607 Satellite System.

All communications between the main computer and the satellite data processors are through the 1607 tape units (13). The program for the satellite data processors must be closely integrated with that of the main computer. Satellite read/write operations are conducted over the same channels that handle data transfer to and from the main computer and tape units; however, satellite selection accomplishes a different mode of data handling within the logic of the tape units. Under normal tape operations, a 48 bit word is gated to the tape units, where it is disassembled into 6 bit bytes, and a parity bit is generated for each byte. The eight characters, each with its parity bit, are then written sequentially onto magnetic tape. After selection for direct transfer to the satellite processor, data transfer is initiated from the main computer exactly as though it were to be written on tape. However, each 48 bit word is disassembled by separate circuitry into 12 bit bytes with no parity bit. These bytes correspond to the word size of the satellite data processor. They are then gated to the satellite, one 48 bit word of output becoming four 12 bit words of input to the satellite. See (13) for a detailed discussion of all 1607 tape unit logical operations.

The satellite communications system utilizes channels 3-4 and channels 5-6 pairs. Read and write control are necessary before initiating direct transfer operations, which effectively means that both channels of the satellite pair must be idle before commencing satellite read or write operations. In the FORTSHARE control system, satellite write operations are used for all forms of computer output, and it is possible for tape operations to be conducted on the same channels as the satellite in use. (Tape units 5-8 and Satellite Station 2, also on channels 5-6). Since all read/write operations of the Fortran 60 Monitor are conducted fully buffered, it has been necessary to effectively unbuffer these operations when the same satellite channel pair is used. This means a penalty of some 20% in time, which is not considered a major drawback. If satellite operations are conducted while tape operations on the main computer are performed on the opposite channel pair, this buffered operation is not disturbed. In any case, when the FORTSHARE control system is used at the main console to process batched jobs, all read and write operations are completely buffered, maintaining the speed of the Fortran 60 Monitor.

The worst case of input-output channel manipulation occurs when a Monitor job with both input and output on channels 5-6 is controlled by Satellite Station 2, also on channels 5-6. When the Monitor Read routine is entered (resident location 3A), the contents of the Monitor Input Buffer is transferred to the resident Read Buffer, and a new record is read into the Monitor Input Buffer.

After exiting to itself, the compiler operates on the record in the resident Read Buffer, and is thus processing one record behind the Monitor Read routine. A similar situation exists with the Monitor Write routine, where the routine first shifts the contents of the resident Write Buffer to the Monitor Output Buffer, then initiates the write operation in buffered form and exits. This action of actually processing one record behind itself permits the use of fully buffered operations with inherent advantages of speed, although the action is accomplished in both cases without the benefit of parity or length checking. A problem in timing occurs with the satellite system in the case of programmed write operations, since this information is first diverted to the satellite station before being written on the output tape. Because of the interaction of the satellite channel pairs in direct transfer, both read and write ready conditions are required, but here we have a case where both are probably active because of the buffered operations of the Monitor. In this case, sensing read and write ready before satellite selection is sufficient to guarantee non-interference. Sensing channels inactive before satellite selection is, in itself, not sufficient, since some 400 microseconds of internal processing must be completed during the write operation after the channel is deactivated before another selection may be safely made. Sensing channels inactive, and then sensing read and write ready, is a solution in this case, but in general can lead to difficulties in initialization should opposite pairs be used for tape operations

and no tape units on the satellite channel pair are physically ready to read and/or write.

An example of the coding used in the initialization of a satellite read or write routine is shown below. This method allows any combination of tape and satellite channel usage:

```
       1ZR2   SLJ(N)          SAU(L+9   )        .
(1)           EXF7(61B)       EXF7(62000B)       .
(2)           EXF7(62000B)    ISK1(100)          .
```

In (1) above, the channel is sensed inactive. If active, a previous selection has been made, and the instruction half exits to wait write ready. When ready, an immediate exit is made from the upper instruction of (2). If the channel is inactive when sensed in the upper instruction of (1) above, an immediate exit is made to the sense write ready in the upper instruction of (2). If write ready, an immediate exit is made. If not write ready, a 560 microsecond delay is taken in the lower instruction of (2) to allow for the 400 odd microseconds of tape unit internal processing before a satellite selection may be safely made. This process is then repeated with the other channel of the satellite pair.

6.2 DD-65 Display Unit at Satellite Station 2.

Limited usage of the DD-65 Display Unit was made in a multi-processor environment in (17). The present usage was the first involving high speed operations, which could be either input or

output from either of two processing units - the main computer or the satellite data processor.

Keyboard 1, the alphanumeric typewriter keyboard of the display unit, is a set of low speed electromechanical devices. When selected for input to the associated data processor, the logical selection is not removed when the input has been completed, as with normal digital equipment manipulation. If the keyboard is again selected for input, the processor is indefinitely delayed when the input is attempted. This limitation is the result of the cumbersome electromechanical keying system used, and had been previously avoided by using another display unit external function code, which had no immediate use in the program to perform a deselect immediately following the input operation. The deselect code previously used was SELECT RADAR RANGE SWITCH (See 14, Chapter 2). The selection of the display unit for memory updating had previously been no problem, since only one source of input could be seen by the display. In the present system environment, it has been necessary to perform a deselect operation on the display after each memory updating to prevent unwanted updating of the display memory when information is passed between the main and satellite processors.

During development, it was discovered that occasional hang up of the main computer was induced by some interaction that could not be explained by communications timing. Investigation revealed a conflict of external function codes between the display and the

1607 satellite system. This deficiency was solved by examining all display external function codes in the satellite communications environment. The external function code SELECT RADAR RANGE SWITCH used for deselecting the display, as previously related, was found to be guilty of intermitttently destroying the satellite write chain. The interaction could not be explained by investigation of the satellite system logic diagrams (19). A substitute deselect external function code has been used with no interference whatever. This code (SELECT RADAR TARGET DATA TO AUXILIARY EQUIPMENT) is a legal code for deselect, but has as yet not been implemented logically in the display unit.

7. Concept of System Usage.

The FORTSHARE Satellite System has improved the problem solving capability of the Fortran 60 system by providing the programmer with the ability to insert the best possible decision element at selected positions in his program - himself. Other features have been added to aid the computer user in arriving at results in substantially shorter periods of time. A brief recap of these various features and their proposed usage will now be given.

7.1 Problem Solving Aids.

The routines CHANGE and SATGRAF provide powerful toools for analysis. CHANGE may be inserted at any point in the program, to allow the problem solver to introduce changes in parameters which could not have been foreseen in pre-analysis. The ability to constantly observe programmed output provides the linkage

for the problem solver to make the necessary decisions. The CHANGE routine may also be used as a pure decision element by setting flag values which have been programmed to determine the program direction at strategic branch points.

Coupled with the ability to observe programmed output is the ability to output results in graphical form to a display device. Program linkage with the CHANGE routine can easily be provided to output selected graphs to magnetic tape for hard copy processing off-line. This provides a major time saving device, in that it is no longer necessary to output a large number of hard copy graphs, each of which is most probably based on unknown results. Any user with the capability of operating the main computer under the Fortran 60 Monitor system should easily increase his capability several fold through use of these features.

7.2 File Maintenance.

Routines FILEIN and FILEOUT provide for the manipulation of programs or information files either under satellite control or main console usage. These routines will enable a user to maintain a private file of programs frequently used with ease of access to any desired program. The use of these routines to manipulate files for educational demonstration should not be overlooked. Programs designed for demonstration in many engineering courses could provide an invaluable visual aid.

Routine EDIT provides a dual capability in that it can be used for making changes and additions to user files, or as a

direct correction device for errors detected during program compilation. Correction in this manner can provide the individual user with great savings in time.

7.3 Hybrid Systems.

By utilizing the analog to digital equipment, analog computers, and digital to analog equipment of the Digital Control Laboratory, hybrid control and analysis may be implemented. The subroutine ANALOG provides the basic link between the 1604 Computer and the analog equipment, enabling a user to have the computational power of the 1604 Computer for analysis or control of analog functions in real time.

Although the satellite system provides fast communications between computers, there is a minimum of 600 microseconds delay for each block transfer of data inherent in the 1607 tape unit satellite system logic. When program execution time is considered also, the maximum data rate for digital to analog is about 1 KC, and for analog to digital, about .5 KC. These speeds assume that no other input/output operations are taking place concurrently. Should higher data rates be desired, it is suggested that the A/D and D/A equipment be modified to be compatible with the 1604 direct transfer channel #7 and circumvent the delay inherent in the satellite system.

8. Conclusions.

The system herein described has been tested in its various modes of operation over a period of six months. The philosophy has remained essentially unchanged throughout the growth of the system from a simple interrupt processor to the present linkage of the problem solver and the 1604 computer. It enables a user-programmer to more closely monitor the operation of complex programs, and to firmly imbed himself in problem analysis by acting as an on-line decision element.

Until the advent of computers capable of accepting and acting on non-analytic concepts, the human, slow but versatile, will have to remain as the guiding influence in matters of abstract thought. Many processes yield to no analytical approach except trial and error. These same processes, in some cases, yield readily to solution, or at least approximate solutions, if the power of the human mind can be coupled effectively with the speed of the present day digital computer. It is toward this end that this thesis project has been directed - more closely integrating human analytical insight and electronic computational power in a team approach to a common goal.

BIBLIOGRAPHY

Part I - Equipment and Technical Manuals

1. 1604 Computer, Volume 1: Description and Operation, Control Data Corporation Publication 031A.

2. 1604 Computer, Volume 2: Principles of Operation, Control Data Corporation Publication 032.

3. 1604 Computer, Volume 3: Maintenance, Control Data Corporation Publication 033B.

4. 1604 Computer, Volume 5: Diagrams, Control Data Corporation Publication 035.

5. Control Data 1604 Computer Programming Manual, Control Data Corporation Publication 167.

6. Fortran System for the Control Data 1604 Computer, Flow Charts, Control Data Corporation Publication 183.

7. Fortran-60 Operations, Control Data Corporation Publication SPD-01, July 1963.

8. 160 Computer, Volume One: Description and Operation, Control Data Corporation Publication 068.

9. 160 Computer, Volume Two: Maintenance, Control Data Corporation Publication 069.

10. 160 Computer, Volume Three: Diagrams, Control Data Corporation Publication 070.

11. Control Data 160 Computer Programming Manual, Control Data Corporation Publication 023B.

12. OSAS - The 160 Assembly System, Control Data Corporation Publication.

13. 1607 Magnetic Tape System, Volume One: Description and Operation, Control Data Corporation Publication 037B.

14. Data Display Model DD-65 Instruction Manual, Data Display Incorporated.

BIBLIOGRAPHY

Part II - Reports and Literature

15. Program Control System for a Satellite Mode Computer System, by R. L. Hogg and D. C. Glover, 21 December 1962, Project Report for Digital Control Laboratory, USNPGS.

16. Control System Programming, Remote Computing, and Data Display, by R. L. Hogg and D. C. Glover, USNPGS Thesis, June 1963.

17. Direct Fortran Operations Utilizing the DD-65 Display Unit, by G. H. Leach and R. J. Simonic, June 1963, Project Report for Digital Control Laboratory, USNPGS.

18. Multiprogramming, by E. F. Codd, Advances in Computers, Volume 3, edited by Alt and Rubinoff, Academic Press, 1962.

19. FORTSHARE Programming Document, by G. H. Leach and A. J. Perrella, USNPGS, May 1964.

20. Human Engineering Guide to Equipment Design, by Morgan, Cook, Chapanis, and Lund, McGraw-Hill 1963. Chapter 2 - "Visual Presentation of Information".

APPENDICES

APPENDIX I - A

1. Identity

Title: FORTSHARE RESIDENT CONTROL SYSTEM

Category: Control System

Programmers: CDC Fortran 60 Version of March 1962
Modified for Satellite Implementation
by G. H. Leach and A. J. Perrella

Organization: U. S. Naval Postgraduate School

Date: March 1964

2. Purpose

This control system integrates all normal Fortran 60 functions with satellite processing station implementation. The system has been developed stressing complete compatibility with normal Fortran 60 operations and general philosophy adaptable to full scale time-sharing. The system presently allows for complete remote control of the main computer from a satellite processing station.

All modification details have been flow charted in Appendix I-B. All system coding may be found in (19). The remainder of Appendix I-A is devoted to detailing the usage of resident routines from running programs or service routines.

3. Permanent Low Core Entries to Resident Routines.

Each low core cell contains the instruction pair

SLJ (N)   SLJ (ROUTINE)   .

These cells are entered via return jumps. The normal jump instruction of the lower half cell looks into an open sub-routine which terminates with an unconditional jump back to the original low core linkage cell, except for certain error exits. These must be mentioned in a normal LOC statement for PROGRAM, MACHINE, or SUBROUTINE use as follows:

```
LOC(EFM = 17)
 .
 .
LDA(IA)       SLJ4(EFM)    .
```

Details of each linkage cell usage are given in the following subsections.

3.1 Cell 11 - BCD READ.

This low core cell provides the linkage for BCD reading of standard Fortran records of 120 character length. It is entered with the mnemonic equipment designator in the A register. On exiting the routine the 120 character record is located in the resident READ BUFFER (R = 100), one character per cell, in the lower six bits.

3.2 Cell 12 - BCD WRITE.

This cell provides the linkage for writing normal Fortran BCD records of 120 character length. On entry, the 120 characters to be written must be in the resident WRITE BUFFER (W=300), one character per cell, in the lower six bits, and the mnemonic equipment designator in the A register.

3.3 Cell 13 - BINARY READ.

This cell links with the normal Fortran binary reading routine. It is entered with the mnemonic equipment designator in the A register, and on exiting one 54 word binary block will be located in the resident read buffer from R to R+53.

3.4 Cell 14 - BINARY WRITE.

This cell provides the linkage to the normal Fortran binary writing sequence. On entry the A register should contain the mnemonic equipment code and the 54 word block to be written should have been positioned in the resident WRITE BUFFER from W to W+53.

3.5 Cell 15 - BACKSPACE.

This cell links with the backspace routine which backspaces the designated tape unit one record. It is entered with the mnemonic equipment code in the A register. The backspace is always a Read Backspace.

3.6 Cell 16 - REWIND.

This cell provides a linkage to the REWIND routine and is entered with the equipment code in the A register.

3.7 Cell 17 - EFMARK.

This low core cell provides linkage with the routine for writing an end of file mark on the equipment designated in the A register on entry.

3.8 Cell 23 - HARDWARE ERROR.

CELL 24 - PROGRAMMING ERROR.

These cells are used primarily by the compiler and are entered with the address of the first cell of a packed BCD message in the A register. Termination of these messages is made with the illegal BCD character OOB. Cell 23 forces a typewriter output while cell 24 prints on the typewriter only if an error medium has not been specified.

3.9 Cell 25 - NORMAL BCD MESSAGE TO TYPEWRITER.

This cell is entered with the address of a packed BCD message in the A register. Under Monitor control, these messages are diverted to the output medium. This entry is useful in service routine programming when no Monitor arguments are set.

3.10 Cell 26 - SATELLITE READ/WRITE.

This cell provides the linkage for the satellite read and write

routines. It is entered with the length of the block to be read or written in the A register, and Q=3 or 4 for Satellite Station 1 read or write respectively or Q=5 or 6 for Satellite Station 2 read or write respectively. The information to be written must have been packed in the SATELLITE READ-WRITE BUFFER (BUF = 600) before entry. After reading, the information read will be located in this buffer. (CAUTION - Remember that one 1604 word is four 160 words.) Maximum buffer length is 80 1604 words (320_{10} 160 words).

3.11 Cell 31 - TYPE MESSAGE AND SENSE FOR ACTION.

This cell is entered with the address of the first cell of a packed typewriter code message in the A register. These messages are also terminated with the character 00B. After the message has been typed, the computer senses the typewriter for operator action. Upon exit from the sense routine, A contains:

1 if Carriage Return has been struck

0 if Space has been struck

-1 if X is typed

Typing a Period will cause an error return to Monitor and hence to Program Control if Monitor is not in control.

3.12 Cell 32 - TYPEWRITER BCD MESSAGE.

This cell is entered with the address of a packed BCD message in the A register. The linkage provides unconditional output to the console typewriter at any time.

3.13 Cell 33 - REWIND WITH INTERLOCK.

This cell provides the linkage to the LOCK routine and is entered with the mnemonic equipment code in the A register. Any legal code may be used. Action is taken only if the argument is an assigned tape unit.

APPENDIX I - B

TYPEWRITER WAIT LOOP MODIFICATIONS

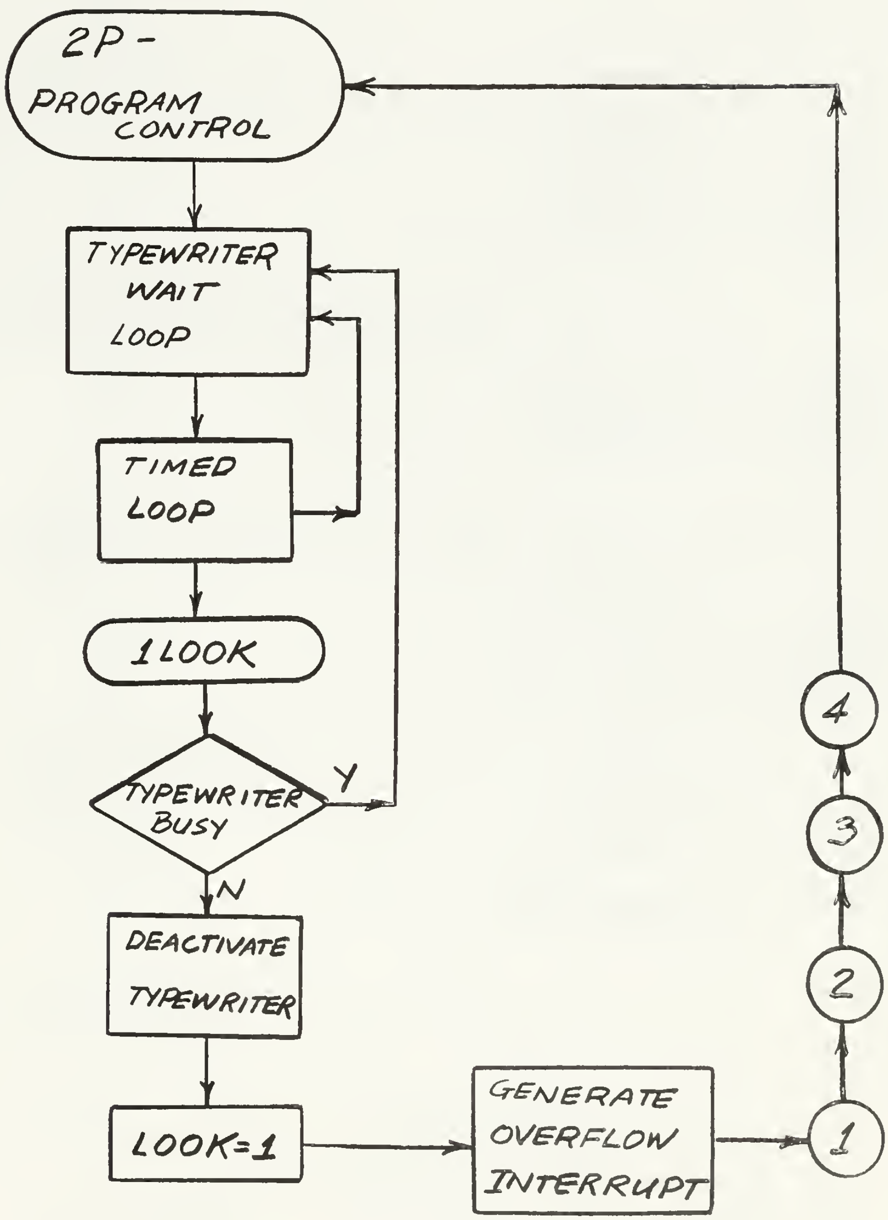


FIGURE I - 1

APPENDIX I - B

INTERRUPT PROCESSOR

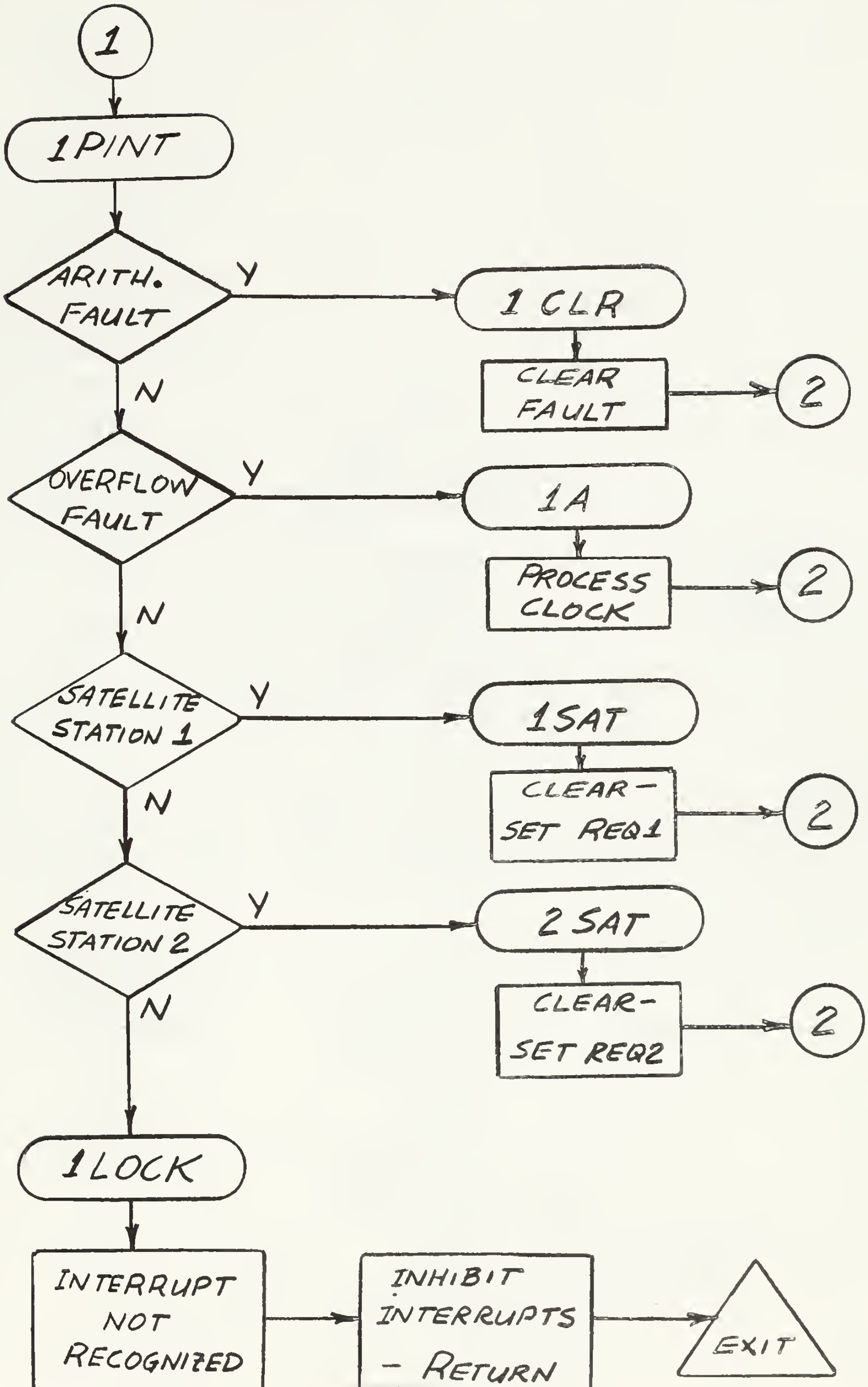


FIGURE I - 2

APPENDIX I - B

EXECUTIVE CONTROL ROUTINE

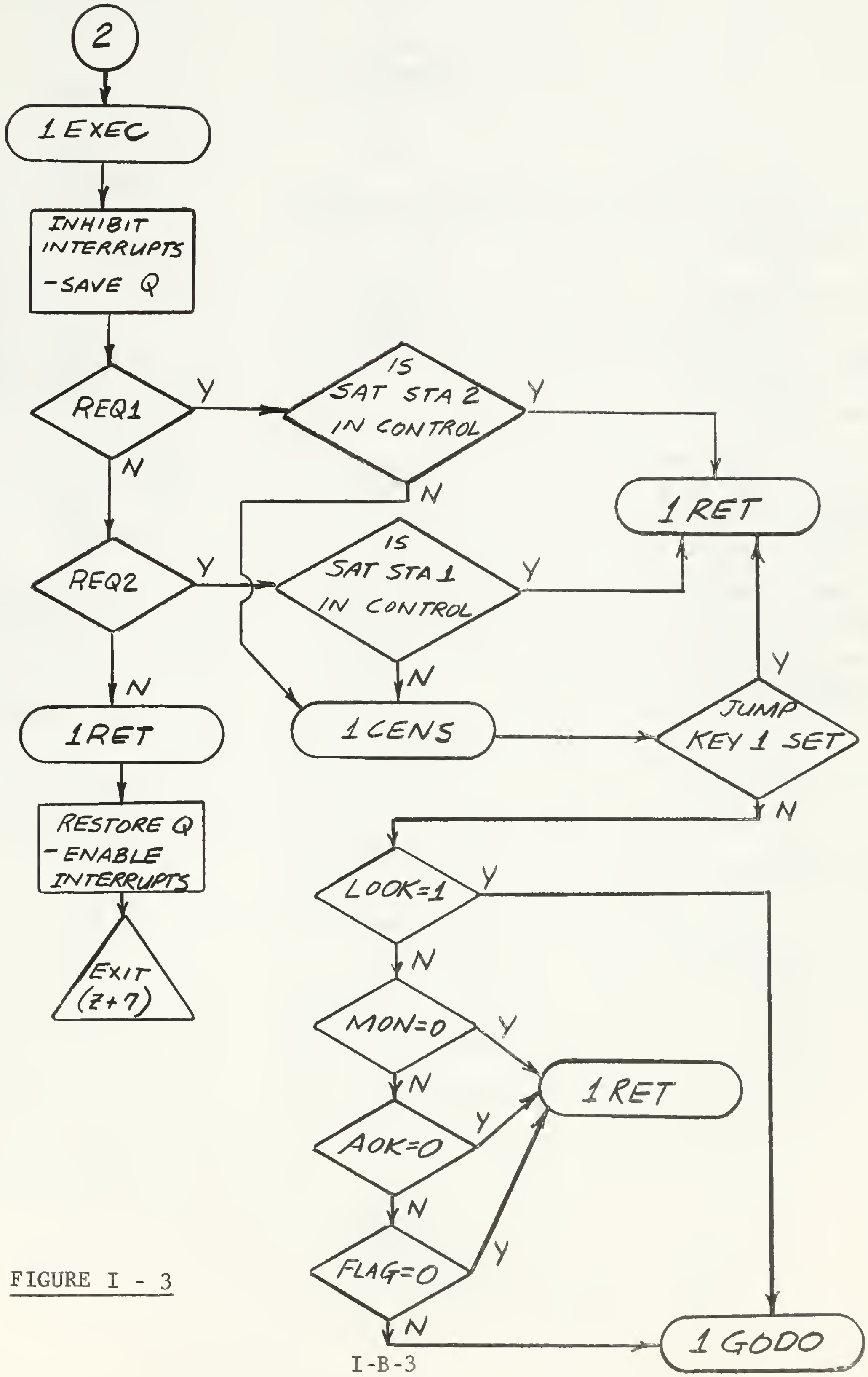


FIGURE I - 3

APPENDIX I - B

SATELLITE REQUEST PROCESSOR

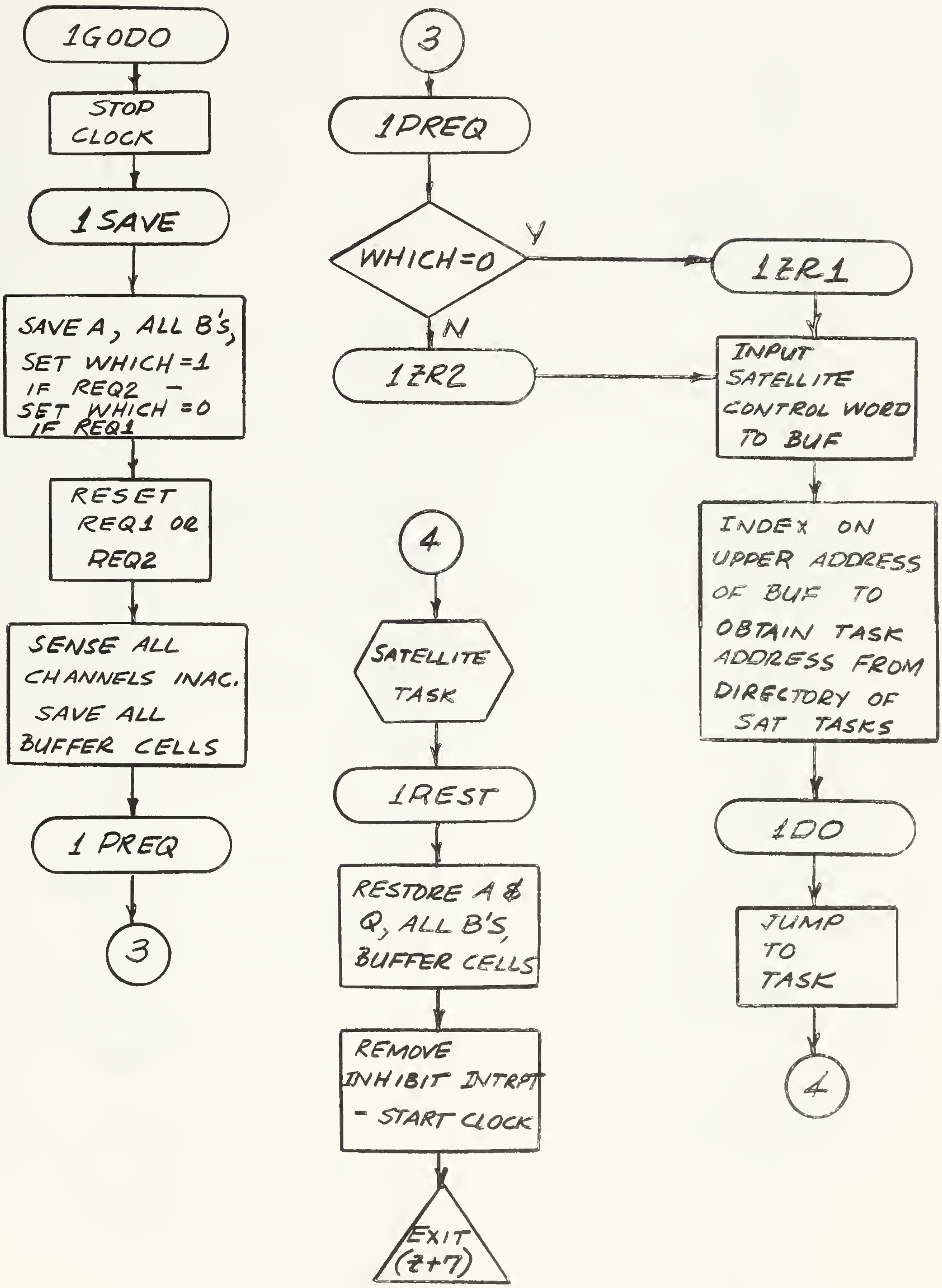


FIGURE I - 4

SATELLITE READ - WRITE ROUTINES

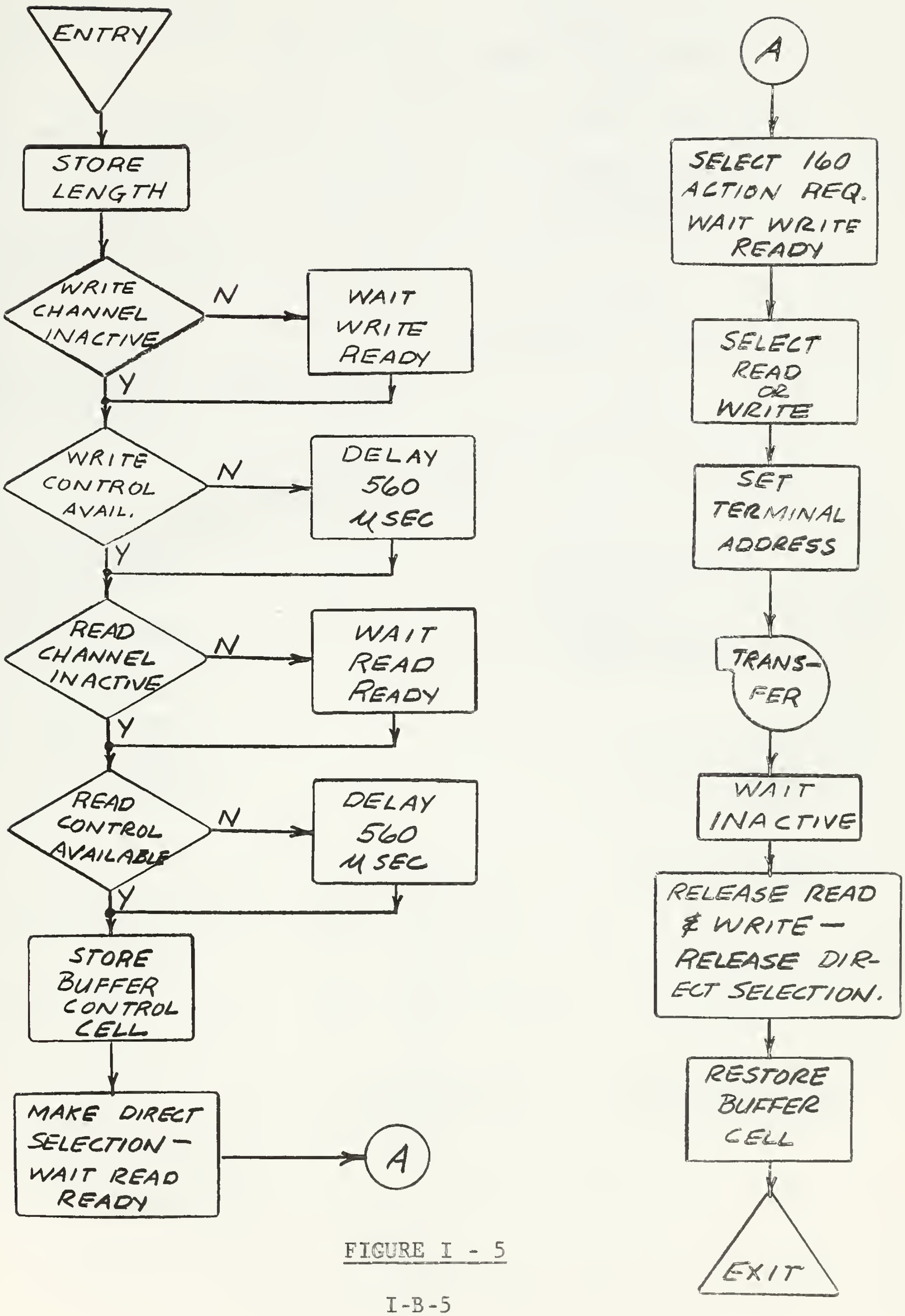


FIGURE I - 5

APPENDIX I - B

SATELLITE CONTROL OF MAIN COMPUTER - INPUT

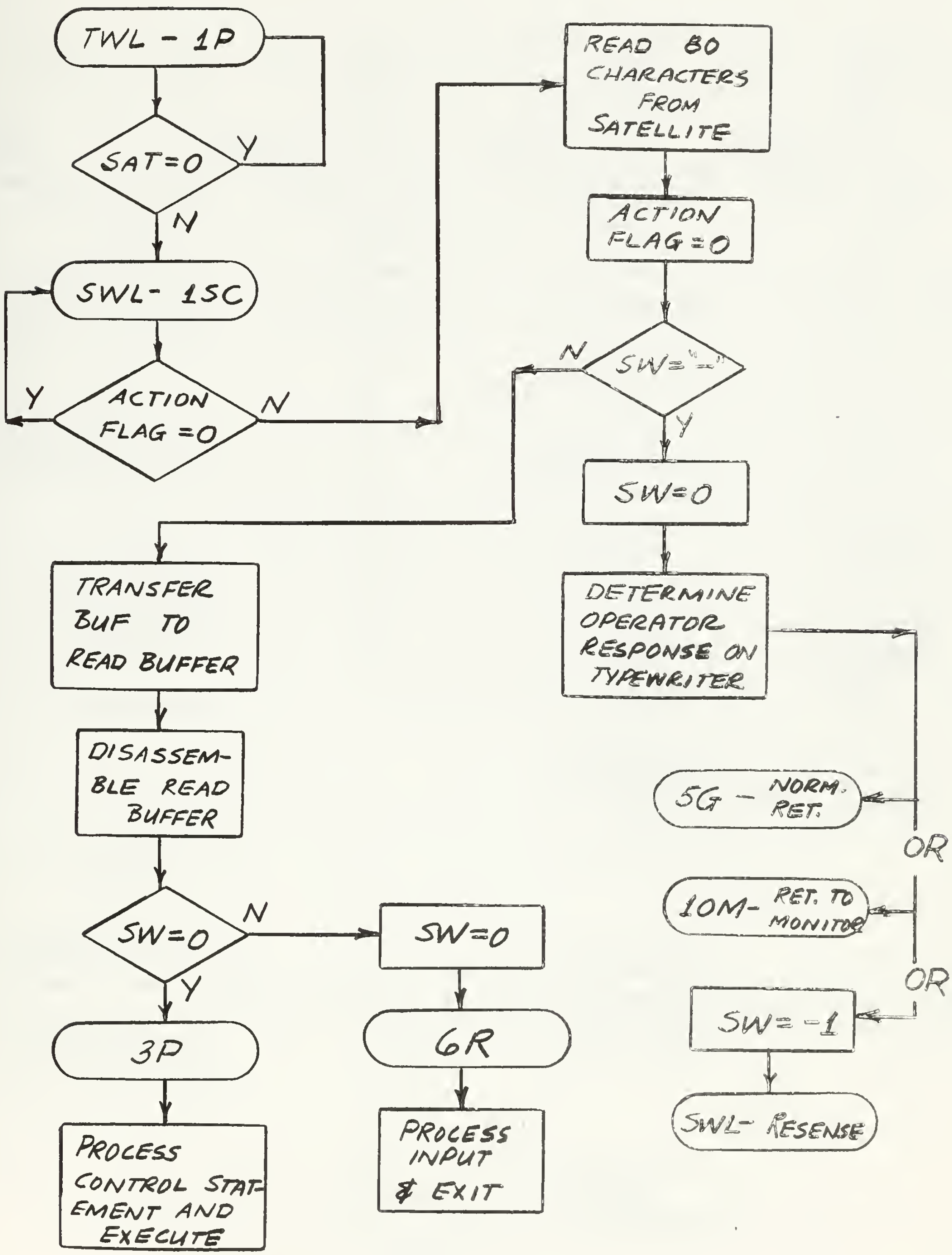


FIGURE I - 6

APPENDIX I - B

SATELLITE CONTROL OF MAIN COMPUTER - OUTPUT

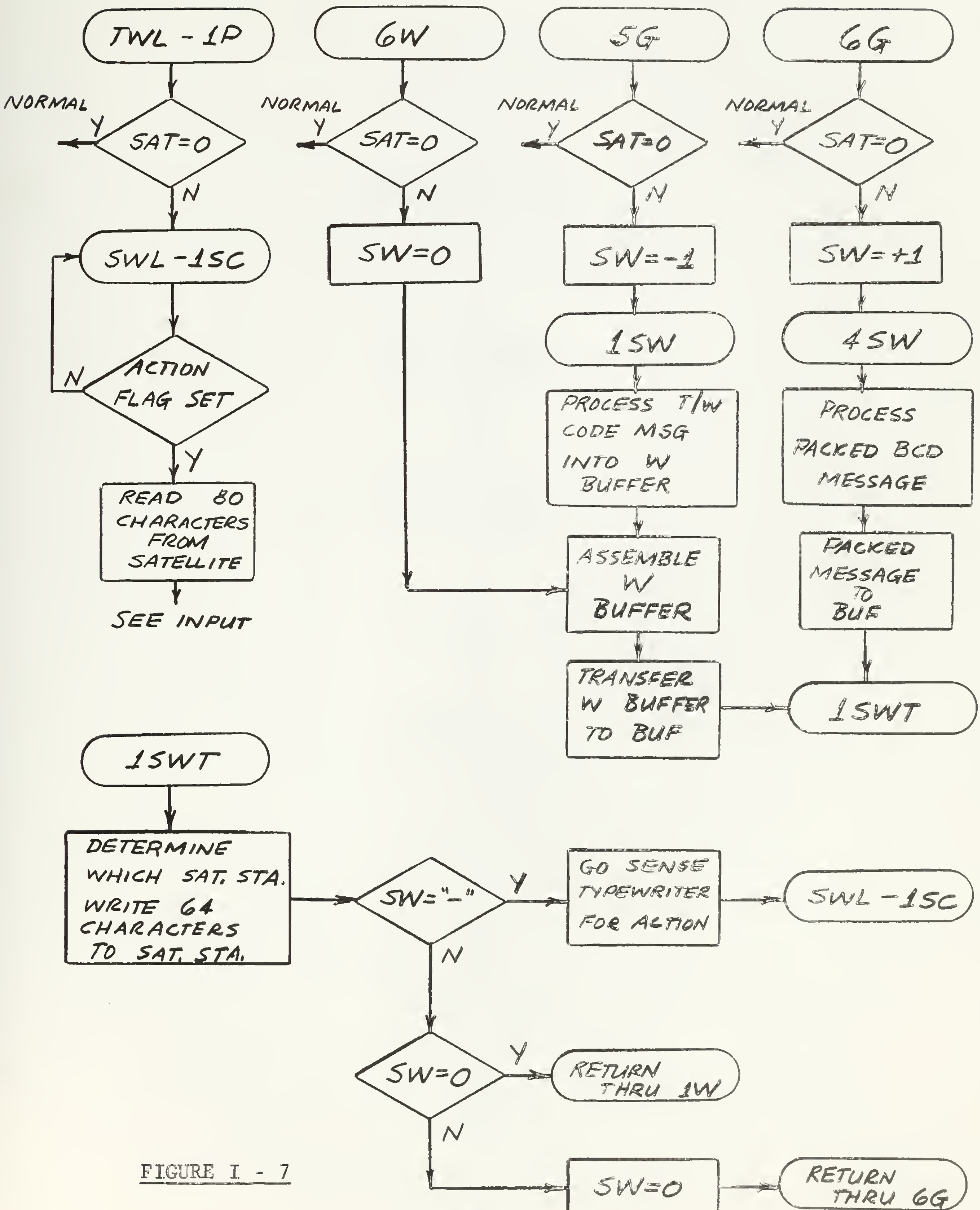


FIGURE I - 7

APPENDIX I - B

BCD OUTPUT TO SATELLITE STATION -

LINE PRINTER SIMULATOR

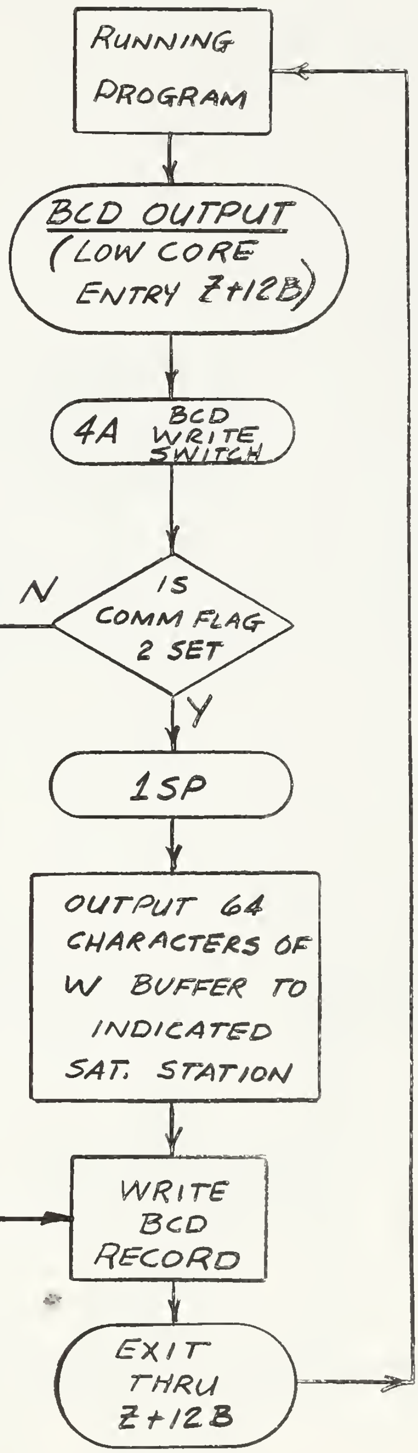


FIGURE I - 8

APPENDIX I - B

TASK 1 - SATELLITE LIBRARY CALL

(INTEGRATED WITH 160 BOOTSTRAP ROUTINE)

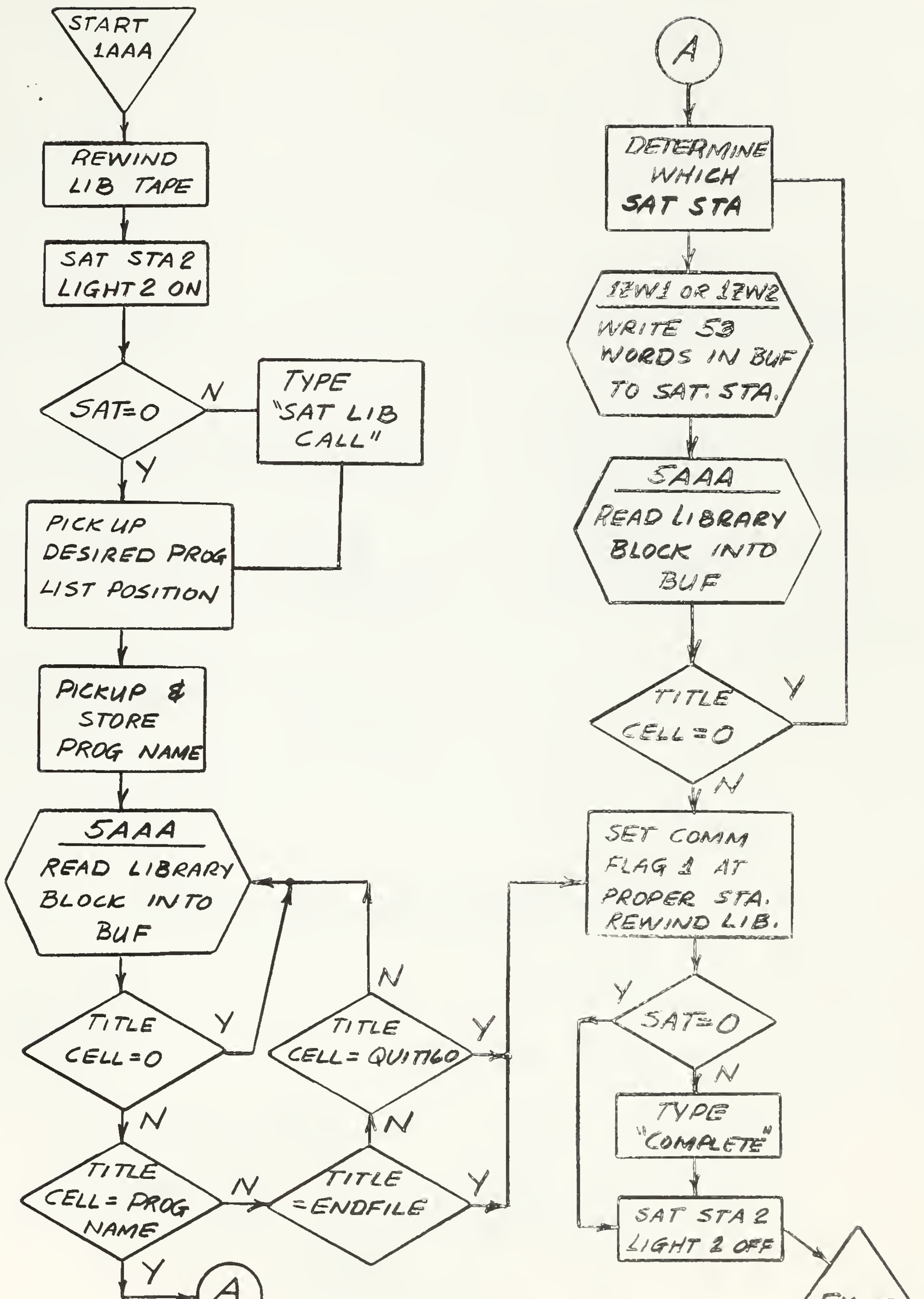
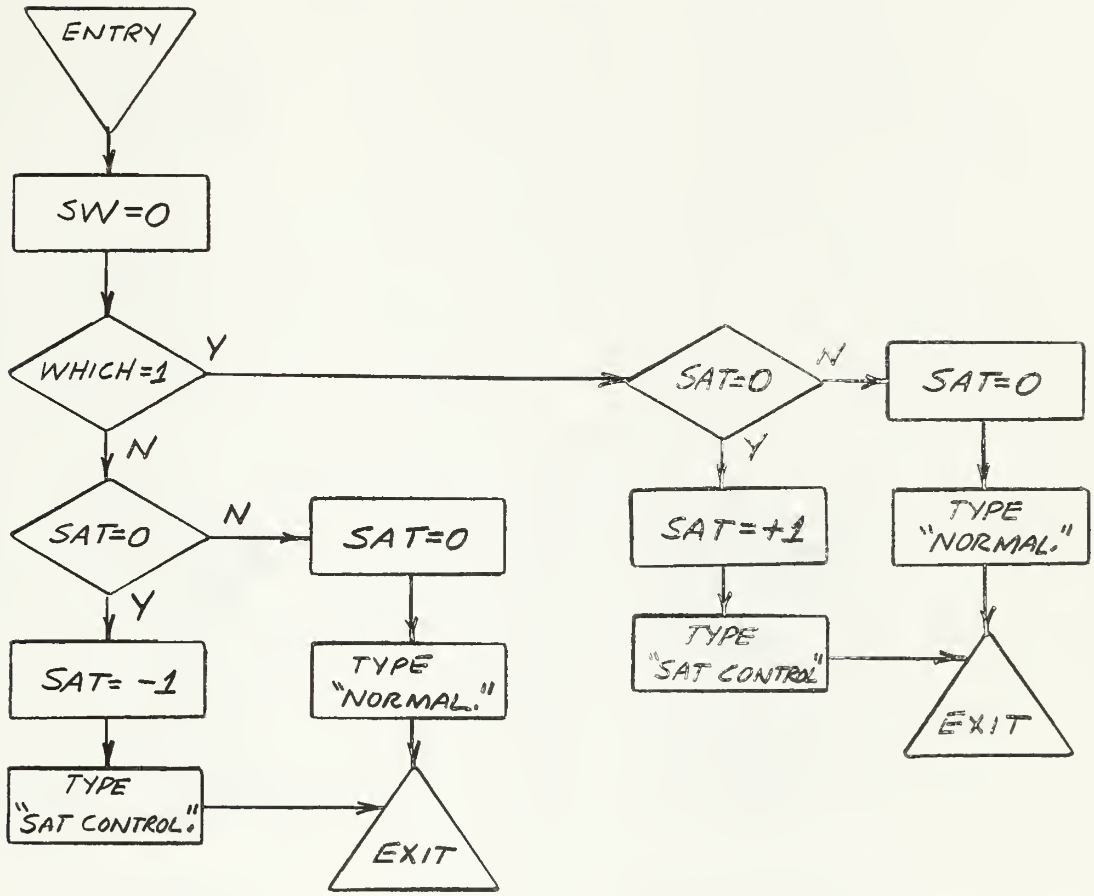


FIGURE I - 9

TASK 2 - SET AND RESET SATELLITE CONTROL SWITCH



TASK 3 - SET SATELLITE ACTION FLAG

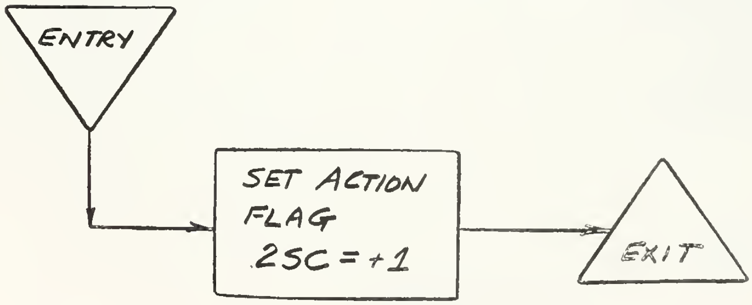


FIGURE I - 10

APPENDIX I - B

TASK 4 - TAPE UNIT STATUS REPORT

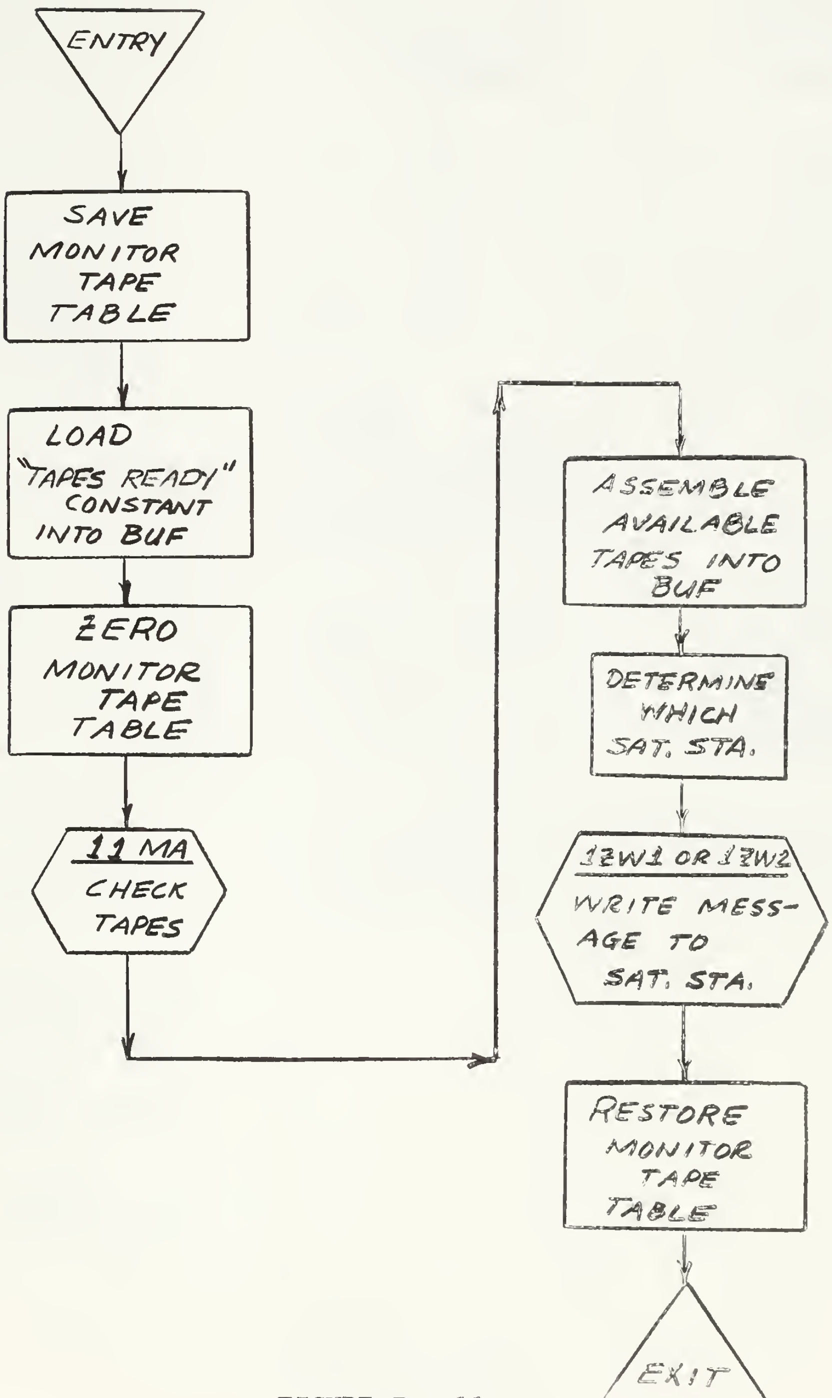


FIGURE I - 11

APPENDIX I - B

CLOCK PROCESSING

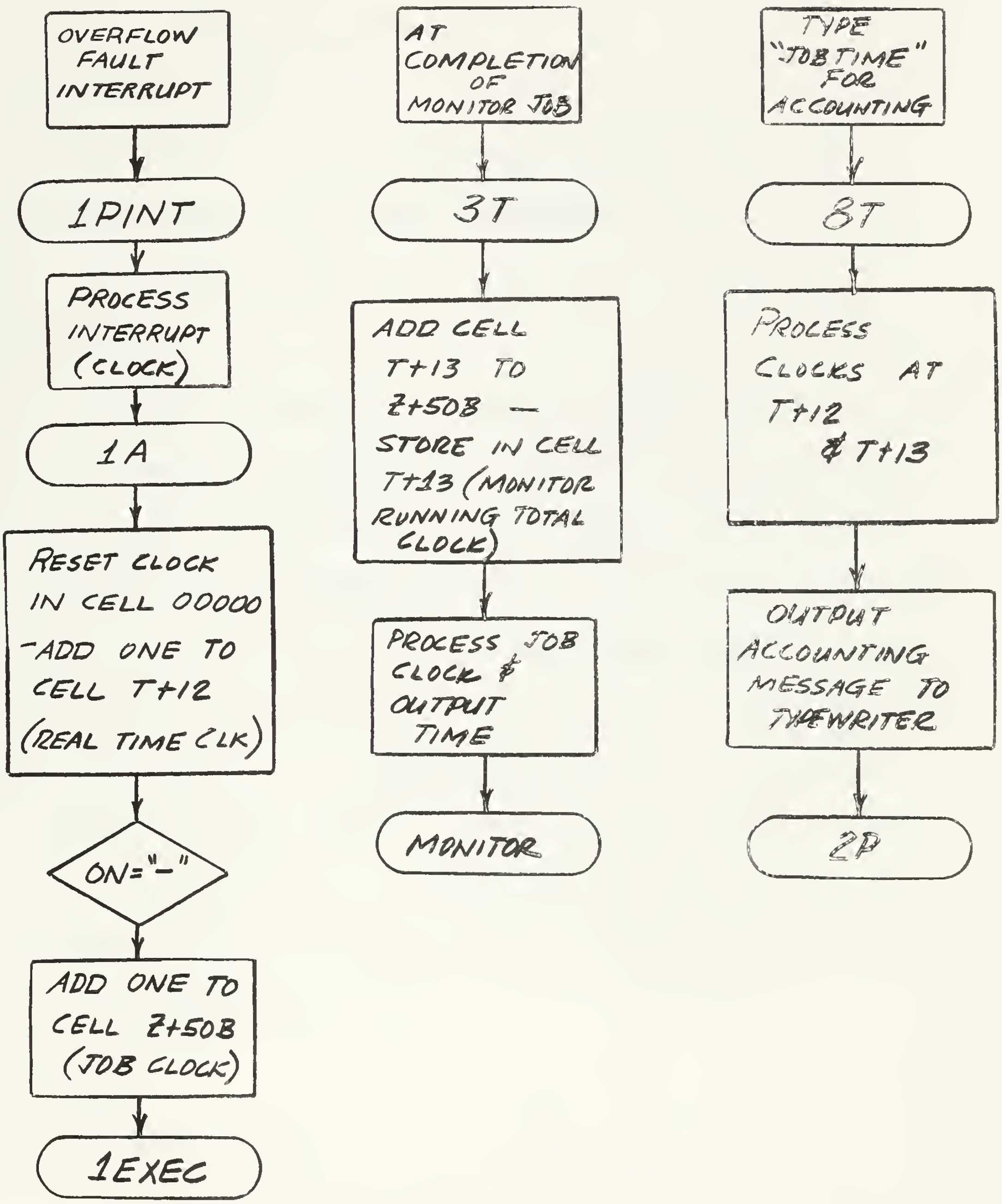


FIGURE I - 12

APPENDIX II - A

1. Identification

Title: SATELLITE CONTROL

Category: 160

Programmers: G. H. Leach and A. J. Perrella

Organization: U. S. Naval Postgraduate School

Date: March, 1964

2. Purpose

This routine provides the integrated control system for satelliting the equipment of the Digital Control Laboratory to the 1604 computer located in the Computer Facility. The routine has been developed using the philosophy of FORTSHARE. It is offered as an example of specialized programming to integrate the functions of special equipment at a satellite station based on the general philosophy of the FORTSHARE resident control system. In the equipment complex located in Satellite Station 2 (the Digital Control Laboratory) the Data Display DD-65 Display Unit is utilized as the control-input-output medium.

3. Usage

3.1 Normal Usage.

3.1.1 The FORTSHARE library should be mounted on tape unit one and the remote control switch located under the main console front should be in the REMOTE position (counter clockwise).

APPENDIX II - A

This energizes the remote partial control panel at Satellite Station 2.

3.1.2 Satellite cables for channel 5-6, located in the rear of the channel 5-6 1607 tape units, should be connected. These are normally connected at all times.

3.1.3 At Satellite Station 2, connect the 160 computer cables, the DD-65 cables, and the satellite cables, at the centralized patch board.

3.1.4 Turn on DD-65 400 cycle power supply. Turn on the DD-65 display unit. CAUTION - Always hold the CLEAR - RUN switch in the CLEAR position when turning the display unit on or off. Place the CLEAR-RUN switch in the RUN position. Place the display mode switch in the 160 ONLY mode.

3.1.5 Load the SATELLITE CONTROL routine either by BOOTSTRAP from the main computer or with paper tape into the 160 computer. Clear and run the 160 computer at location 0000. The satellite complex is now ready to exercise control over the main computer.

3.1.6 Restarting the 160 program at location 0000 at any time will reset all lights and internal flags to their position before stopping. Clearing and running the program at location 0001 will reset all flags and lights to the quiescent condition.

3.1.7 The remote partial control console may be used as desired for control of the main computer.

3.2 Detailed Functioning.

3.2.1 Display Unit Keyboard 1.

3.2.1.1 The extended alphanumeric keyboard of the display unit is referred to as Keyboard 1. It is used exactly as an electric typewriter As each key is struck, the corresponding letter, symbol, or number will be displayed on the bottom line of the presentation on the right hand display tube.

3.2.1.2 The right hand tube is used for printed material of an input, output, or control nature. The presentation of the right hand tube is that of a 16 line rolling page. Pressing the carriage return key increments the screen presentation upward one line in the same manner as an electric typewriter or a high speed line printer. When the screen is filled, the top line is deleted from the presentation when the screen is incremented.

3.2.1.3 All of the keys are represented by standard BCD codes. Special functions have been assigned the BLANK key and the TAB key. The BLANK key clears the bottom line of the display and is useful if correction is necessary and the operator does not desire to increment the page with the carriage return (CR). The TAB key is cyclical in operation and provides tabs to space

7, space 24, and space 41, for compatability with the authors use of Symbolic Machine Language.

3.2.1.4 A control statement typed onto the screen is treated exactly as if it had been typed on the console typewriter except that the period does not trigger the main computer. With the satellite control system, the OUTPUT key of keyboard 2 is pressed to send the typed statement to the main computer. See the following section.

3.2.2 Display Unit Keyboard 2.

3.2.2.1 The 20 keys to the left of Keyboard 1 and the 10 keys above Keyboard 1 are referred to as Keyboard 2. These keys output a characteristic BCD code when pressed and sampled by the associated computer. They are used to select certain sub-routinized functions for the control system. The composition of this keyboard is shown in Figure II-1.

3.2.2.2 Keyboard 2 is used for many system control functions. Those necessary to control will be discussed in the following sections, while those keys which are specialized in nature will be discussed in other appropriate appendices.

3.2.2.2.1 SAT CONTROL - This key performs the function of taking control of the main computer at the satellite station. When pressed, the main computer is interrupted and a message is

delivered to the main console that a satellite station has control. This deactivates the main console typewriter and the main computer is then sensitive to the display unit keyboard as its control medium. The MAIN CONTROL and SAT CONTROL lights in the P register of the 160 computer also indicate when the console typewriter and the satellite complex have control, respectively.

3.2.2.2 LINE PRINT - This key activates the main computer switch which diverts all BCD output from the main computer to the satellite station before it is written on a designated output medium. Each 120 character line is displayed as a line on the right hand tube. A new line from the main computer automatically increments the page. The size of the characters on the screen have constrained the line width because of readability from a normal sitting position in front of the display, and only 62 characters are displayed. This is of no consequence for normal Fortran statements, but restricts computer output presentation if the entire 120 character field is used. Users may very easily limit their output format to 62 characters. This feature may be used at any time.

3.2.2.3 TAPE UNIT STATUS - This key instructs the main computer to examine its tape units and send a message to the satellite relating which are ready to be read. The status interrogation

routine examines READ READY status only and consequently a tape without a write ring will be indicated as ready. This key may be pressed at any time.

3.2.2.2.4 MASTER COMM FLAG - This key enables the key to its right, the COMM FLAG SET key. This lockout protective feature is for operator protection, since the condition of the communications flag in question determines the interpretation given to input from the main computer.

3.2.2.2.5 ERROR - This key serves as the reset medium for removing system error indications. All functions of Keyboard 2 are closely interlocked and any function which is not legal at the moment is locked out. Pressing the illegal key causes the ERROR light to be lighted. Press the ERROR key to turn out the light and proceed.

3.2.2.2.6 RESET - This key resets all control flags used in the satellite control system. Its purpose is for easy resynchronization of the main computer and the satellite system whenever stoppage of the main computer requires that resident must be reread. This condition zeros all internal flags in the main computer, and if a Satellite Station had been in control, would reverse this condition at the main computer. The satellite system must be reset to coincide with the main computer before beginning again.

APPENDIX II - A

3.2.2.2.7 COMM FLAG SET - This is a signal device which is enabled by the MASTER COMM FLAG key. When the MASTER COMM FLAG key is lighted, this key can be used to signal the display graphing routines (see Appendix V) to continue. When the MASTER COMM FLAG key is not lighted, pressing this key causes an error indication.

3.2.2.2.8 FLEX INPUT - This key is the control key for entering Fortran programs via the satellite data processor for processing at the main computer. See Appendix XI.

3.2.2.2.9 SLOWDOWN - This key may be used at any time to slow the presentation on the right screen. When the key is pressed, the associated light is lighted, and a delay of .4 seconds is inserted between each line of BCD input from the main computer. This permits the operator to slow the presentation if desired for inspection. Pressing again removes this selection.

3.2.2.2.10 STOP - The STOP key is used to stop the presentation of all material from the main computer to permit the operator to closely inspect the screen presentation. When pressed, the associated light is lighted and all satellite action ceases. The main computer must then wait with the satellite computer. To allow the program to continue, repress the key. Action will restart and the light will be turned out. Pressing this key when Satellite Control is not in effect will cause an error indication.

Pressing any key except the STOP key when the STOP light is lighted causes no action. The STOP key must be repressed before continuing.

3.2.2.2.11 CLEAR - The CLEAR key causes the memory of the display unit to be cleared and hence both right and left screens will be cleared.

3.2.2.2.12 OUTPUT - This key is the central control key of the entire Keyboard 2 complex. It serves the same function as the period when using the main console typewriter. With the satellite system, the entire control statement as composed on one line of the display is sent in one byte to the main computer for processing, rather than processing character by character and executing on a typed period, as with the main console electric typewriter. The OUTPUT key functions to deliver the control statement to the main computer. Control statements are used in an identical manner to that of the main console typewriter, with only one exception. In the event of typewriter messages to indicate length, parity, or end of file error, operator response is slightly different. All responses to sense typewriter for operator action are as follows:

Type character "3" - equivalent to carriage return

Type character "2" - equivalent to space

Type character "1" - equivalent to X

Type character "." - equivalent to "."

This key is disabled when Satellite Control (SAT CONTROL) is not in effect, and if pressed at this time will cause an error indication.

3.2.2.2.13 All other keys of the group above Keyboard 1 have been assigned no functions as yet, and if pressed at any time will merely cause an error indication. The remaining keys of Keyboard 2 are specialized in nature and not required for job control of the Satellite Station. These keys will be discussed in Appendix III.

3.3 Miscellaneous Aspects of System Usage.

3.3.1 After having read the previous key functions, any operator who can intelligently use the computer at the main console should be able to increase his effectiveness several fold when controlling his jobs from Satellite Station 2 and making use of the system aids provided in Appendix IV and Appendix V. The various functions of the keyboards have been interlocked and protected to the extent that few gross errors are possible. As with any equipment, a bit of practice and experimentation resolves many questions and builds confidence.

3.3.2 The electromechanical keys of Keyboard 1 are prone to sticking and rebound errors. Keys which stick may be released by pressing the button under the console directly beneath Keyboard

APPENDIX II - A

1. Rebound errors cause a double input to the display and are immediately recognizable by two characters appearing for a key strike. Retype the line in this case. This malfunction may be kept to a minimum by keyboard maintenance.

3.3.3 Press all Keyboard 2 keys firmly but gently. If no action is forthcoming, press again. It is not good practice to punch these keys. Bear in mind that some functions require a finite amount of time and that nothing the operator can do will hurry them.

3.3.4 Under normal monitor processing, the output tape is rewound to the unload position when the job is complete. The input tape is not rewound. This feature is valuable for stacked job processing at the main console, but causes the loss of the output tape to the satellite system user. The system now leaves both output and input positioned at the completion of a monitor job under satellite control. The output tape may then be reused for additional program output from its present position. Each complete grouping of output will then be separated by an end of file mark.

APPENDIX II - A

KEYBOARD 2 FUNCTIONS - DD-65

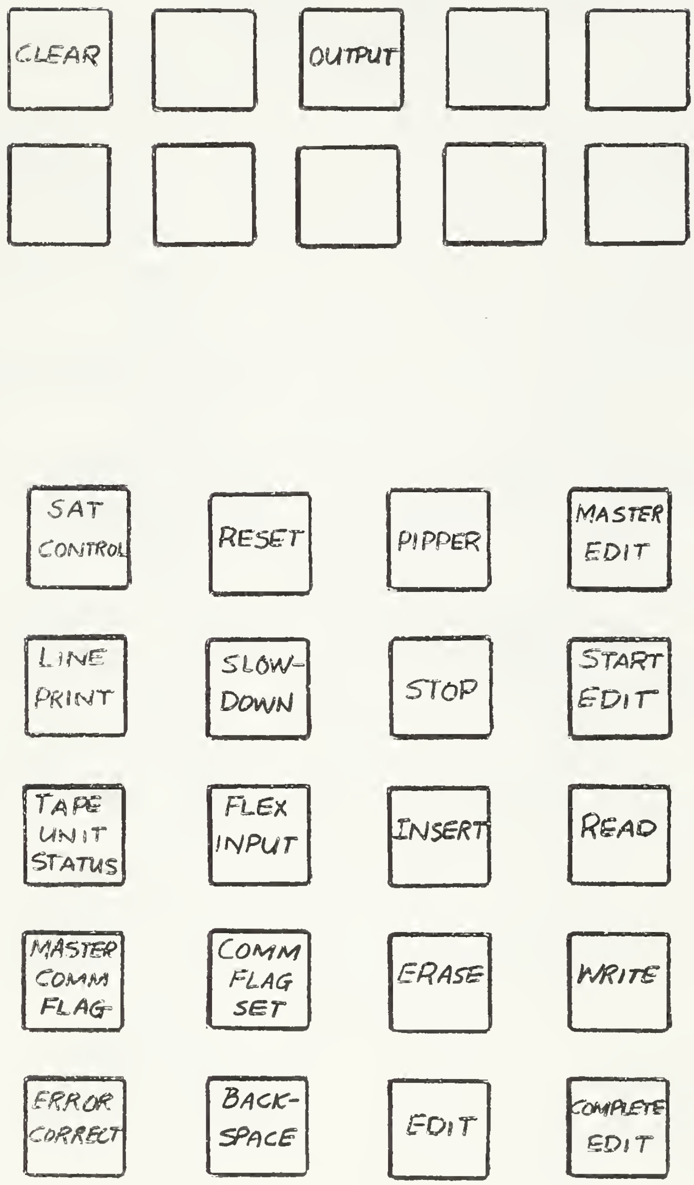


FIGURE II - 1

APPENDIX II - B

SATELLITE CONTROL - CHART 1

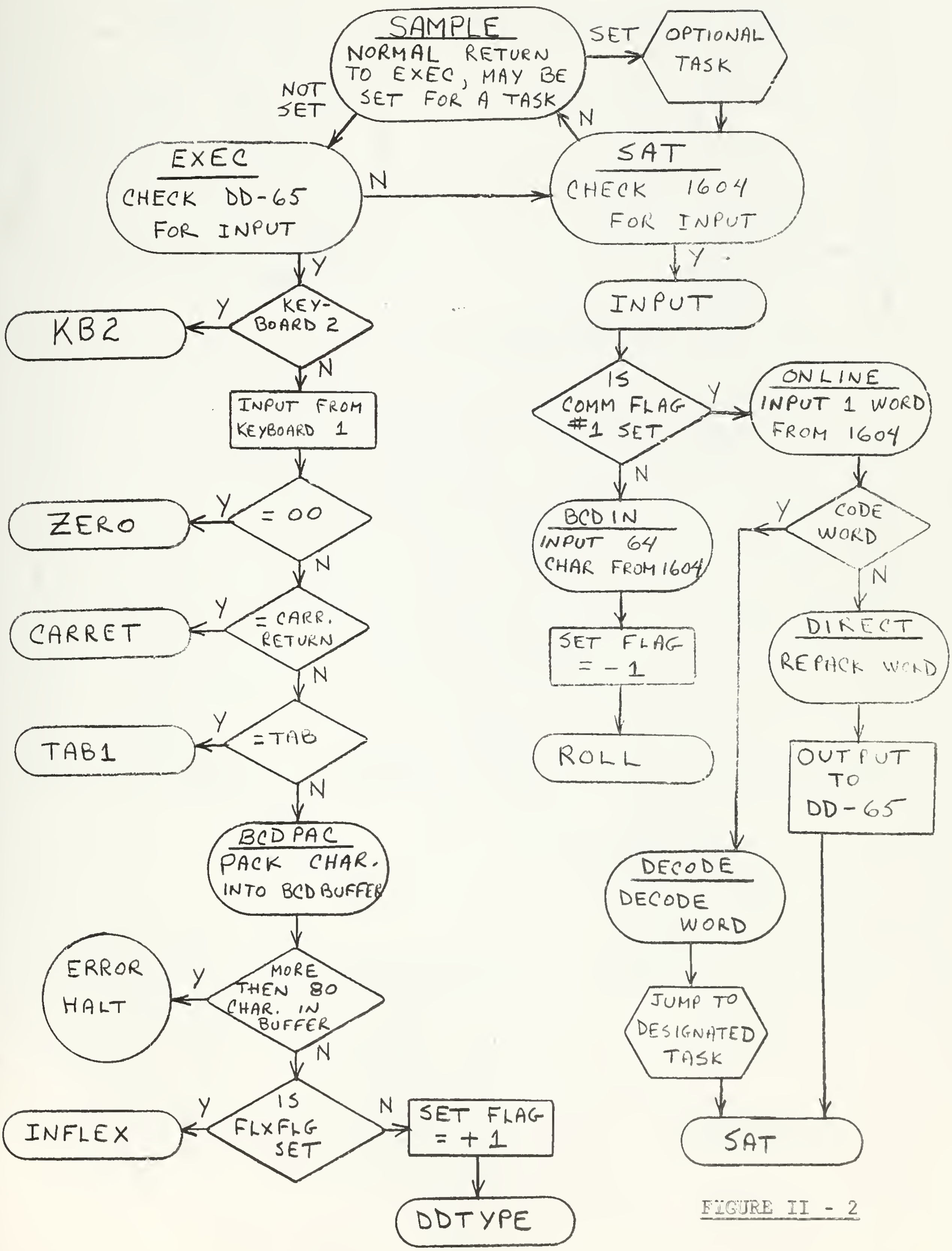


FIGURE II - 2

SATELLITE CONTROL - CHART 2

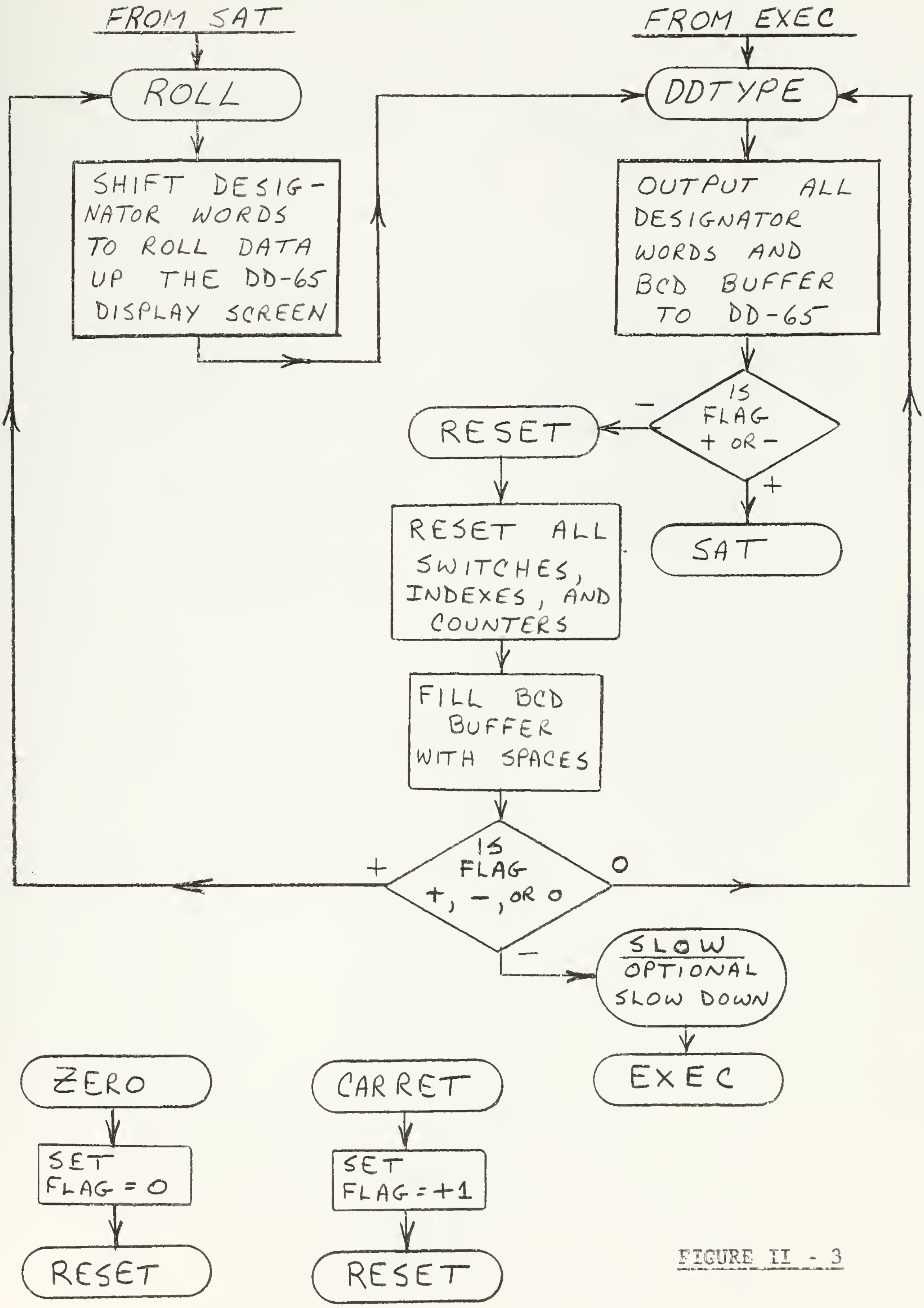


FIGURE II - 3

SATELLITE CONTROL - CHART 3

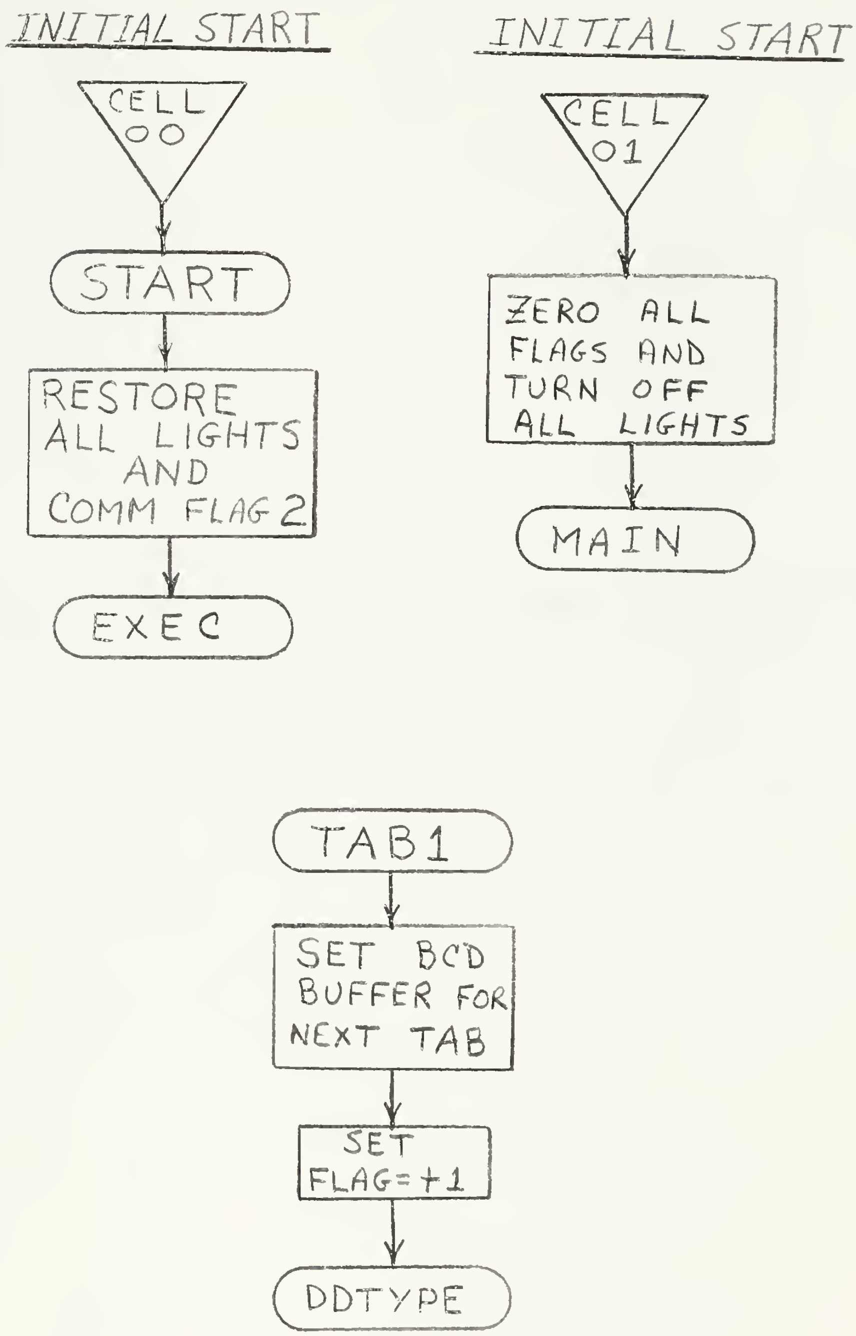


FIGURE II - 4

APPENDIX II - B

SATELLITE CONTROL - CHART 4

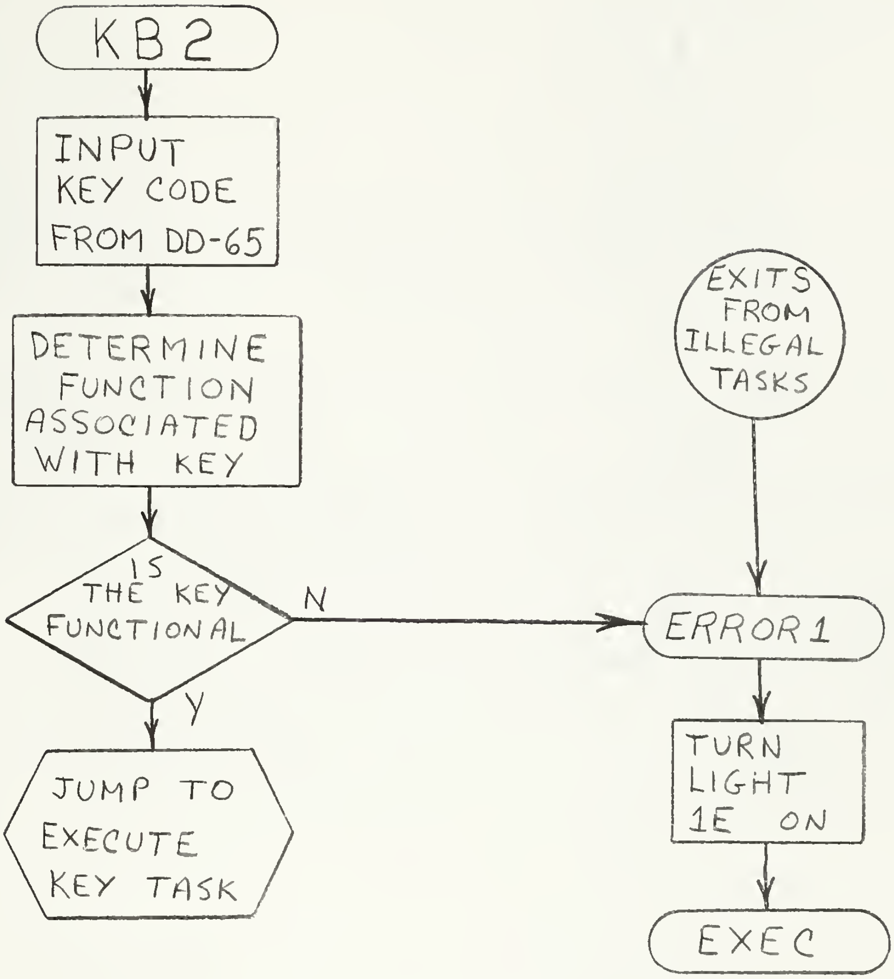


FIGURE II - 5

APPENDIX II - B

SATELLITE CONTROL - CHART 5

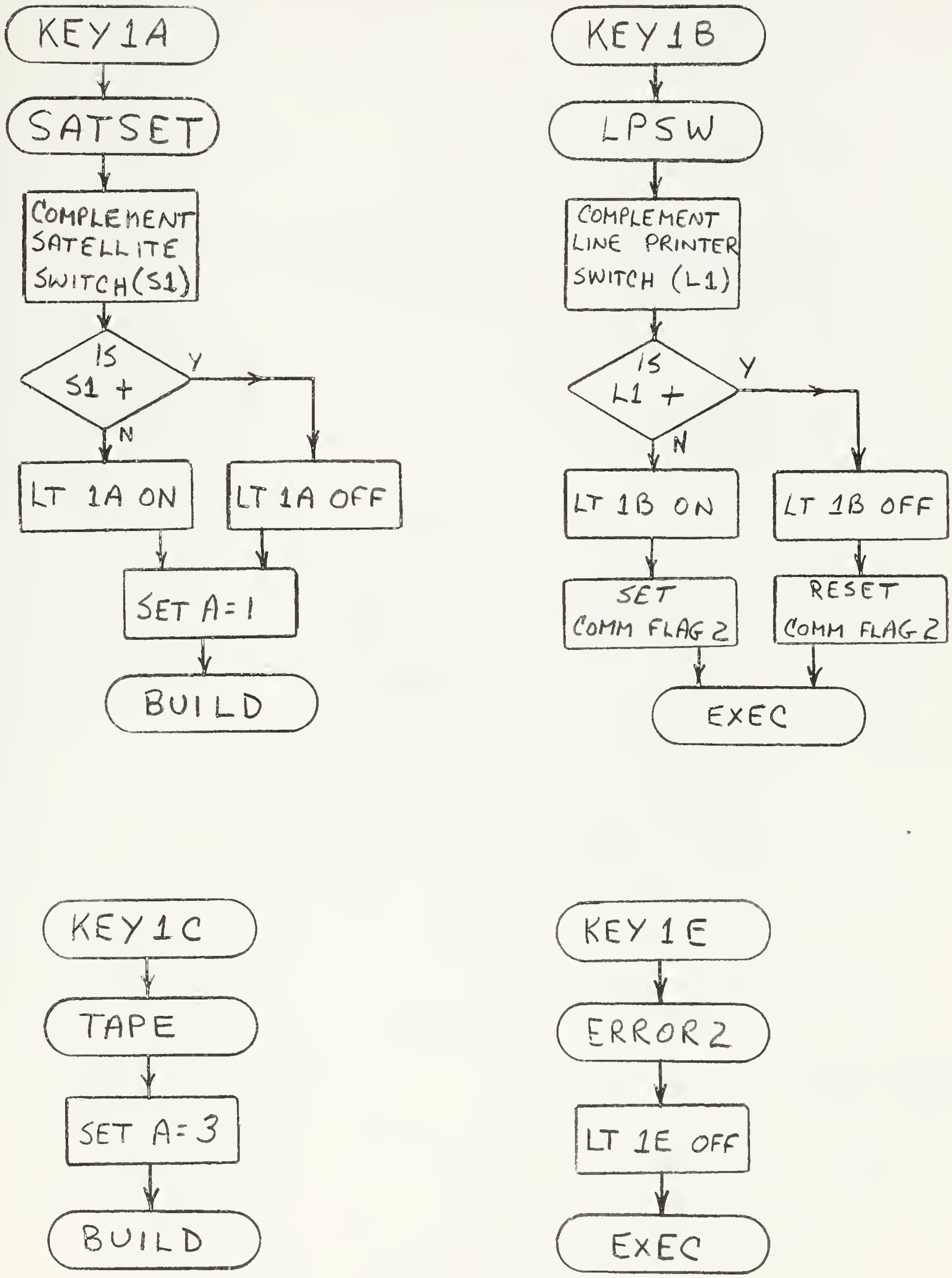


FIGURE II - 6

SATELLITE CONTROL - CHART 6

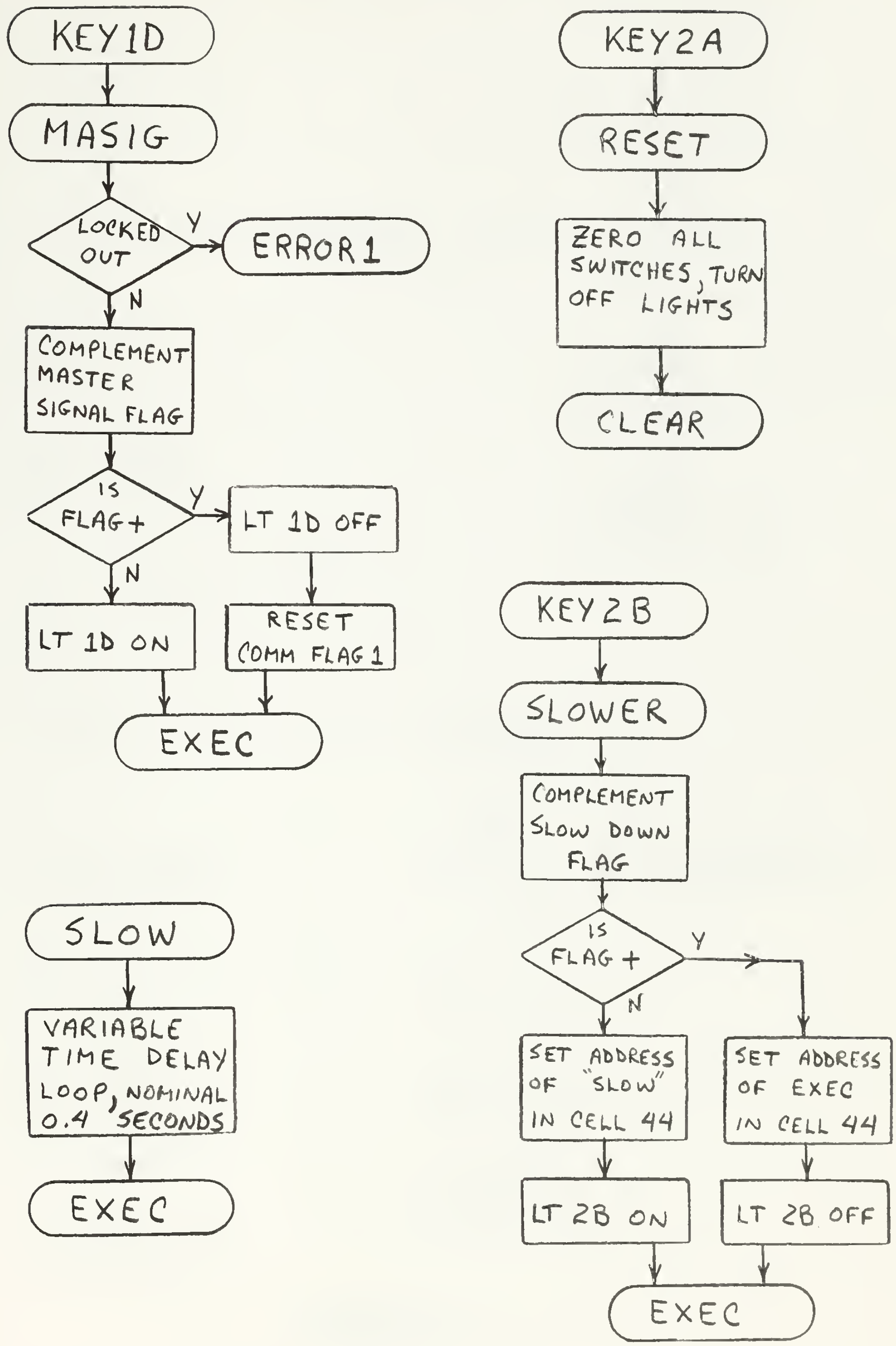


FIGURE II - 7

APPENDIX II - B

SATELLITE CONTROL - CHART 7

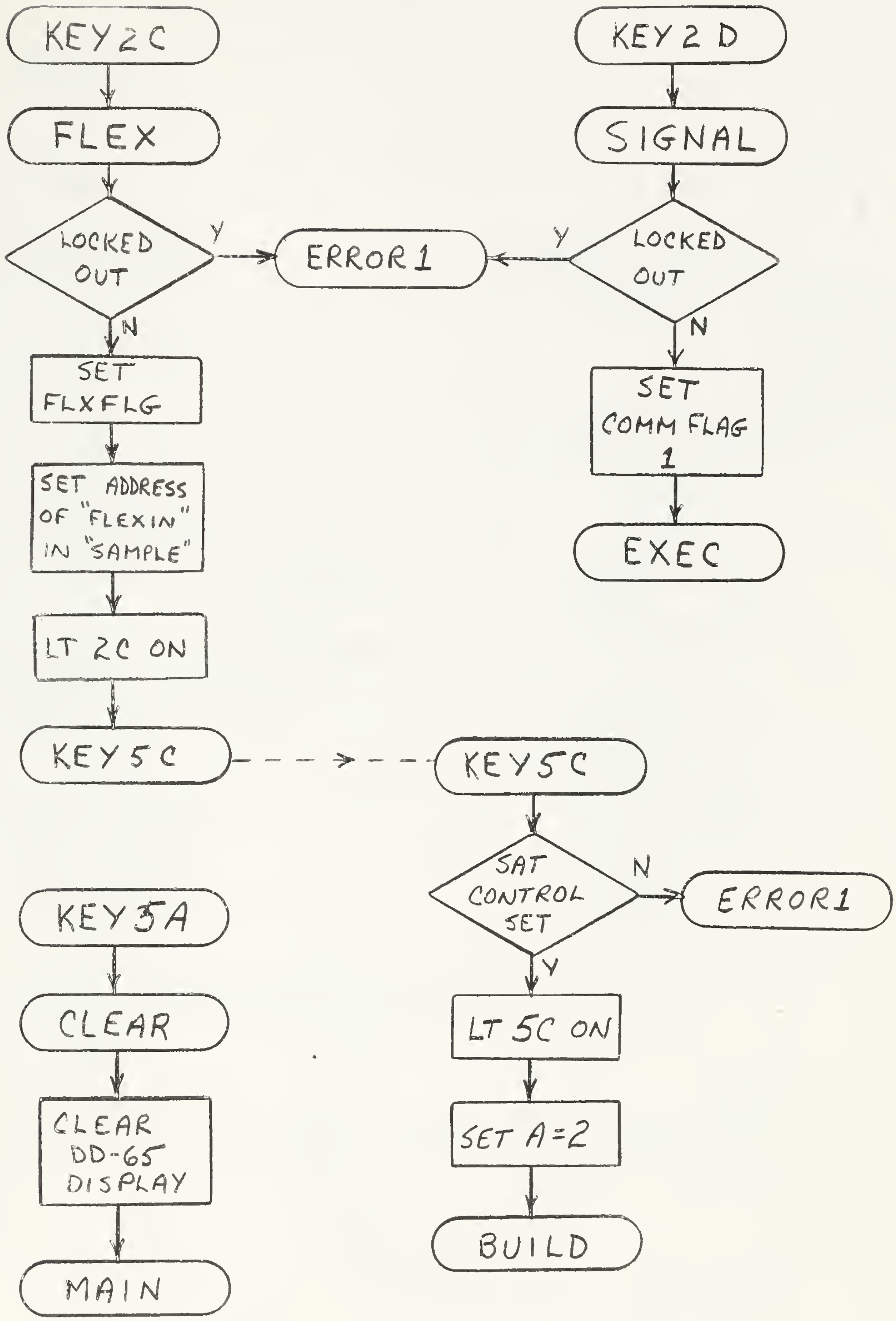


FIGURE II - 8

SATELLITE CONTROL - CHART 8

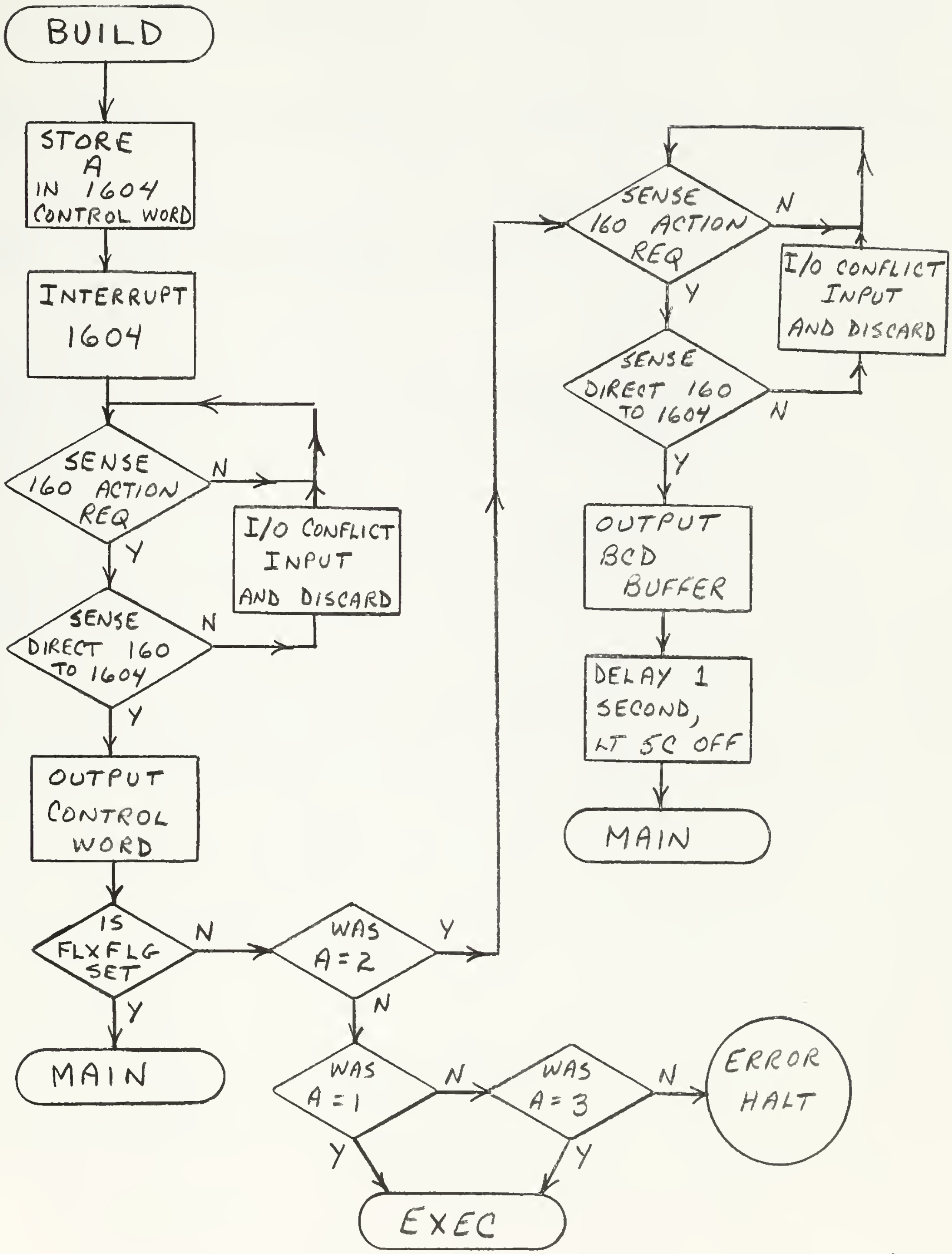


FIGURE II - 9

SATELLITE CONTROL - CHART 9

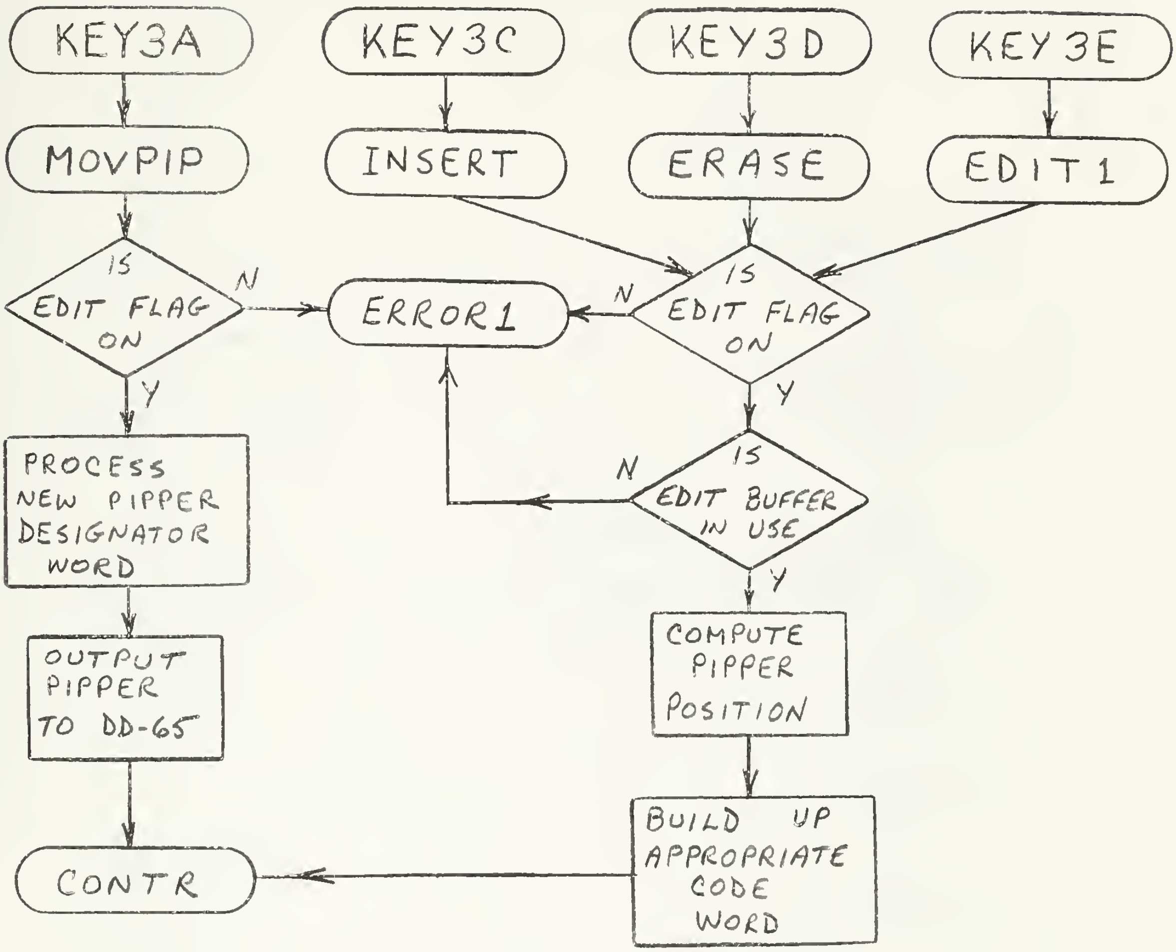


FIGURE II - 10

APPENDIX II - B

SATELLITE CONTROL - CHART 10

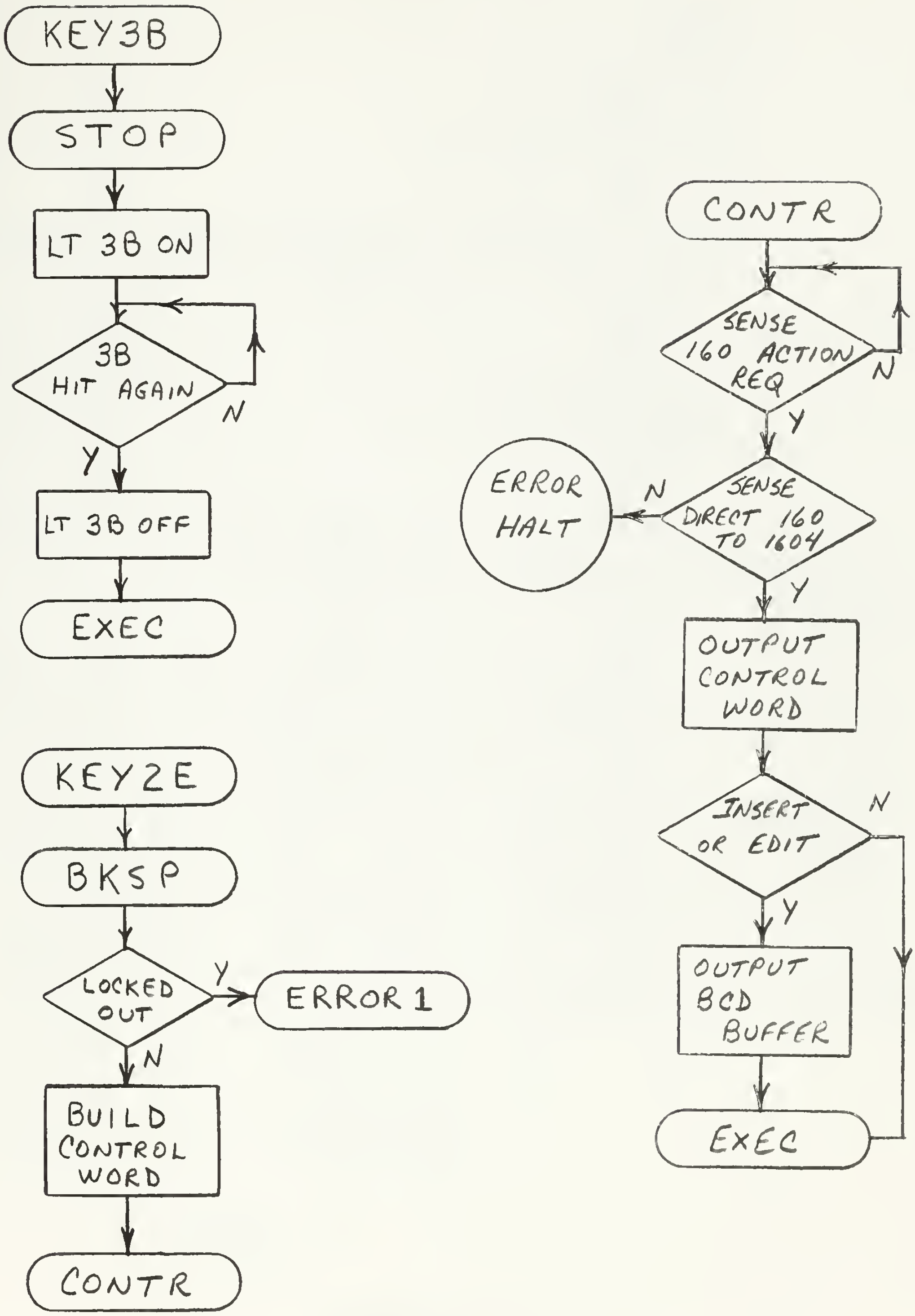


FIGURE II - 11

APPENDIX II - B

SATELLITE CONTROL - CHART 11

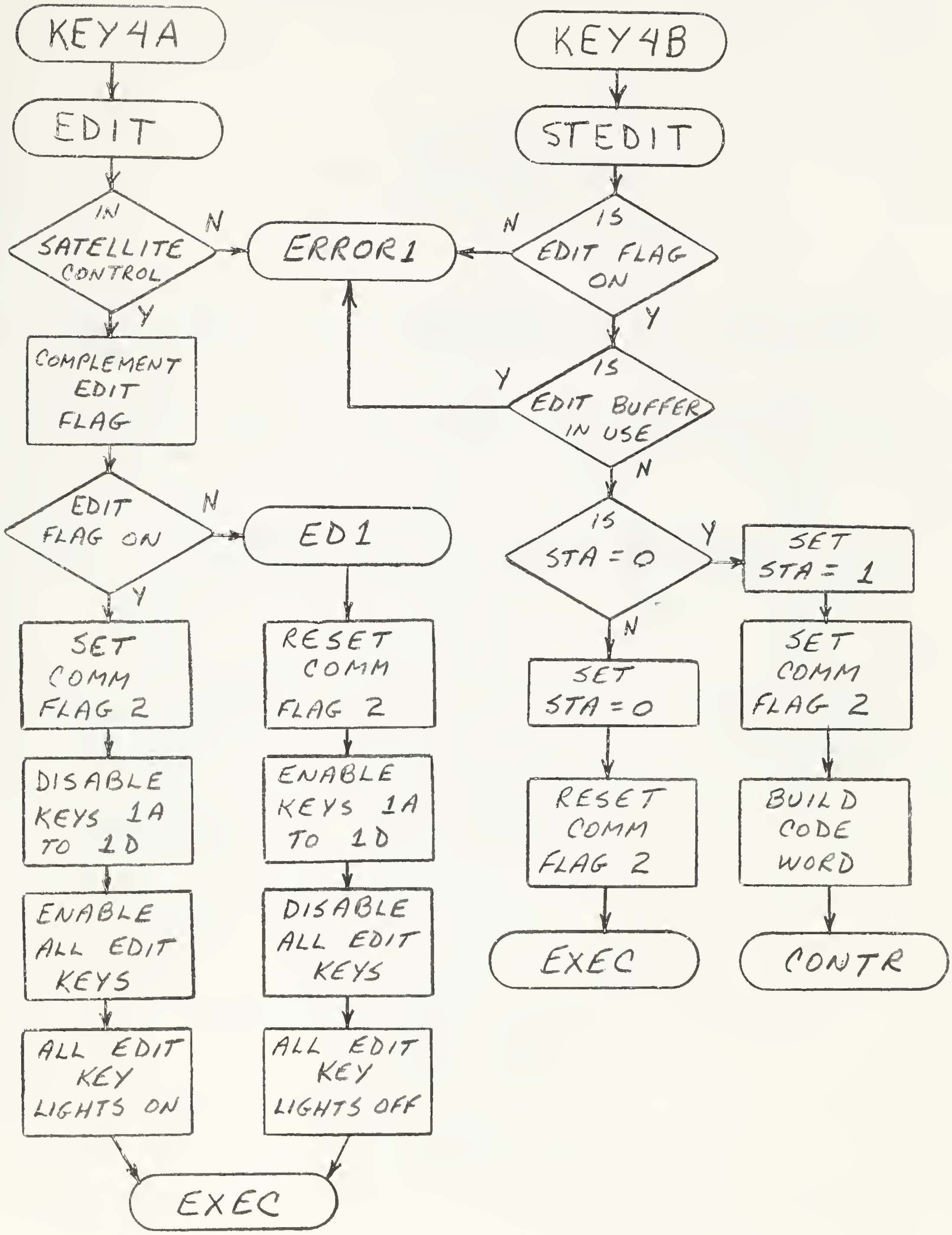


FIGURE II - 12

APPENDIX II - B

SATELLITE CONTROL - CHART 12

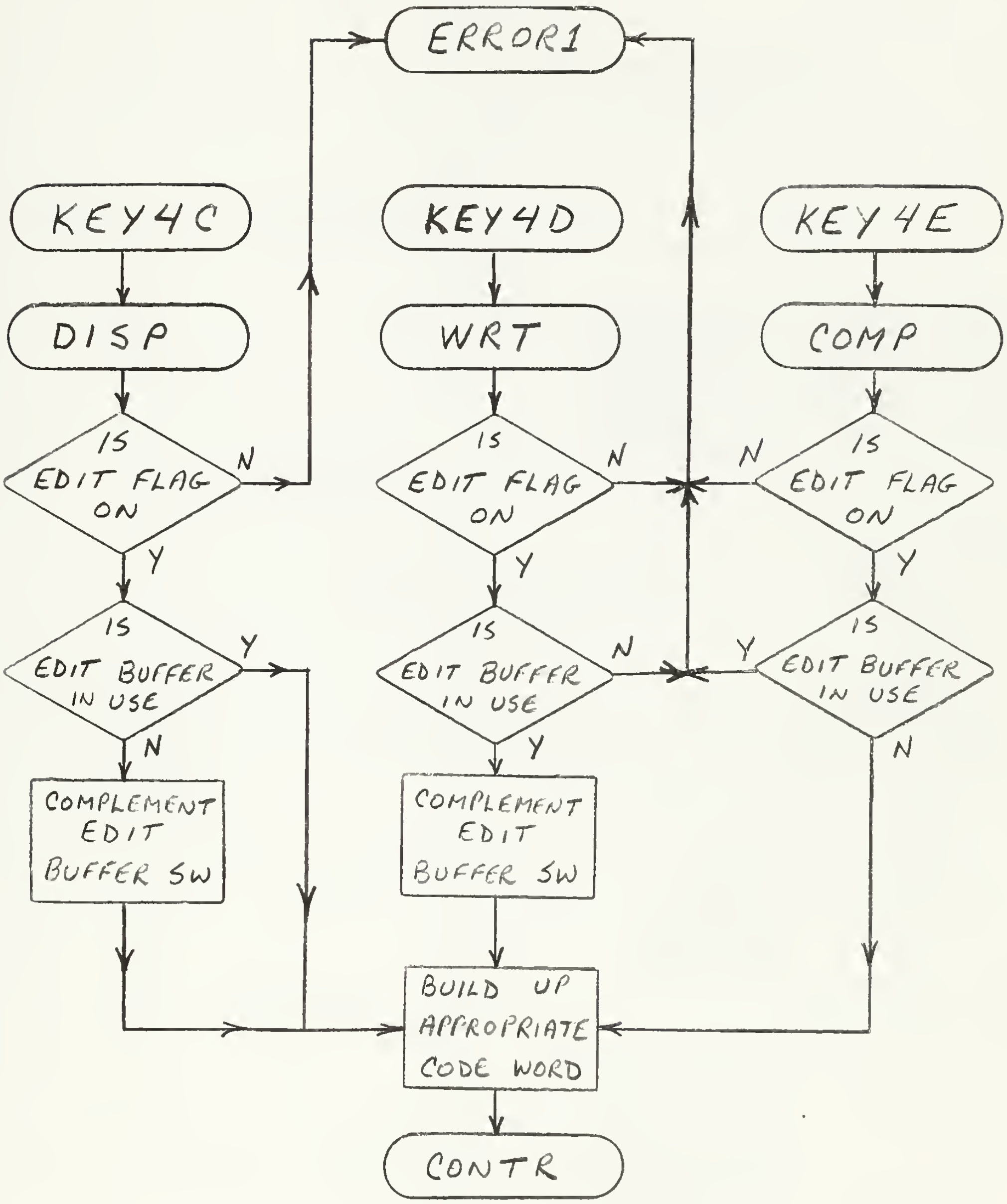


FIGURE II - 13

APPENDIX II - B

SATELLITE CONTROL - CHART 13

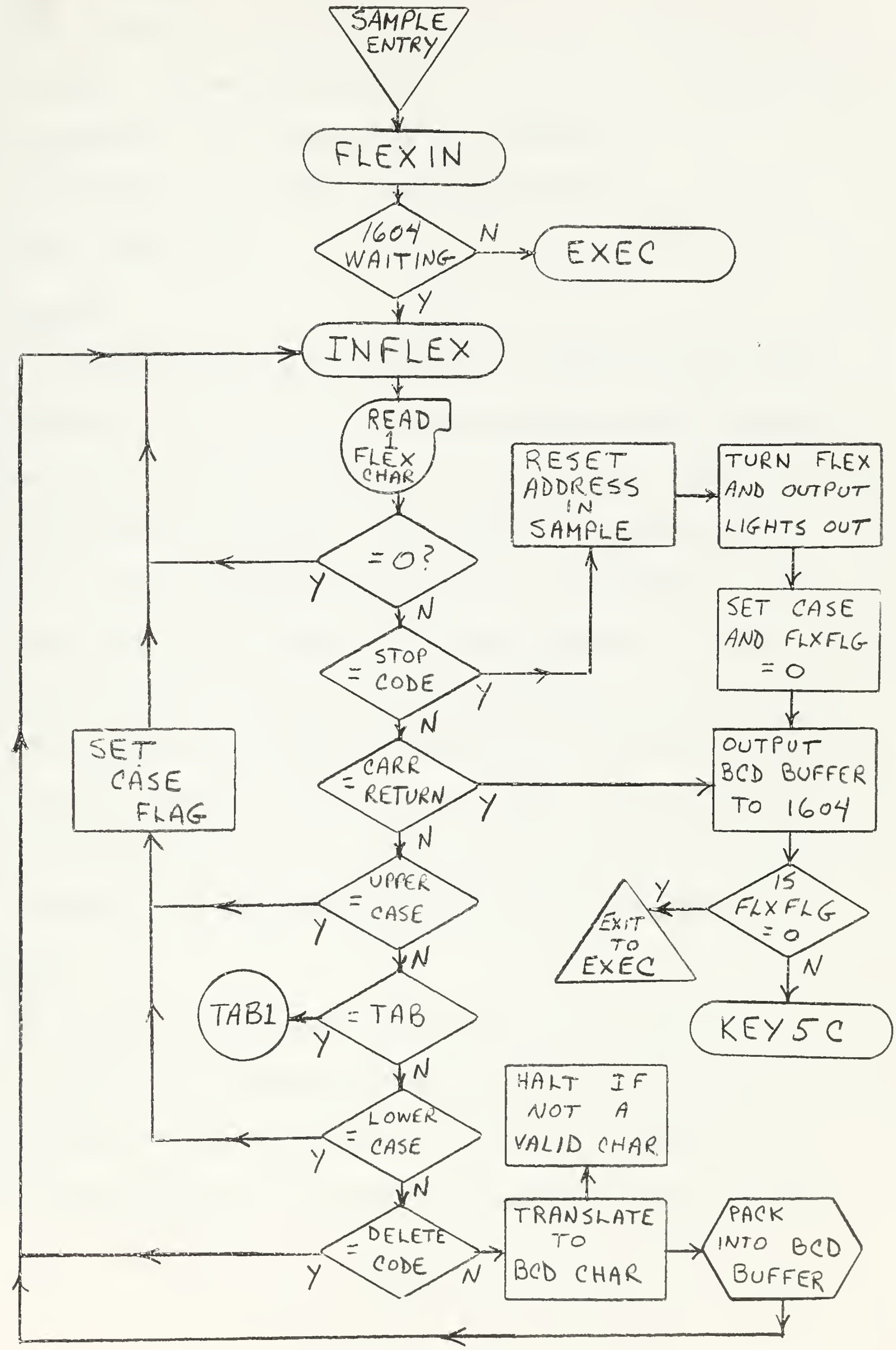


FIGURE II - 14

APPENDIX III - A

1. Identification

Title: EDIT

Category: Q (Service Routine)

Programmers: G. H. Leach and A. J. Perrella

Organization: U. S. Naval Postgraduate School

Date: April 1964

2. Purpose

This routine provides the capability of editing tapes of 120 character BCD records at a satellite control station utilizing the DD-65 Display Unit as the control mechanism. Several modes of operation are provided. The routine is reasonably fast, but it should be emphasized that it is not intended for usage during times of high usage of the main computer. It does, however, provide a means of correcting minor errors detected during compilation without the necessity of removing the tape from the computer for off-line processing. It is also valuable for correction of very large programs where much time is consumed in retaping cards for minor errors or additions.

3. Usage

The routine is typewriter callable and will be called automatically with FORTSHARE when the special call SR is used. An attempt to use the routine at the console typewriter of the main computer will result in an appropriate error message.

APPENDIX III - A

The keyboard of the display console is used as the console typewriter with the constraints enumerated in Appendix II. Special functions of certain keys of Keyboard 2 have been assigned for usage by the Edit routine.

3.1 Special Key Functions.

3.1.1 MASTER EDIT Key.

This key enables all edit key functions, sets the necessary sateellite communications flag, and lights all keys used with the edit routine. It also disables all control keys of column one except the ERROR key (see Figure II-1). Satellite control must be in effect to use the edit routine keys. Keyboard 1 functions normally at all times.

3.1.2 START EDIT Key.

This key is used to commence the edit routine after display of the tape to be edited has started.

3.1.3 READ Key.

This key signals the routine to read 15 records from the input tape into the edit buffer. These records will be displayed on the right tube of the display unit. The records may now be processed as desired. With records in the edit buffer, further pressing of this key will only reread the buffer for another display.

3.1.4 WRITE Key.

This key signals the routine to write the contents of the edit buffer on the output tape. It is selectively disabled when no records are in the edit buffer.

3.1.5 COMPLETE EDIT Key.

This key signals the routine to complete the edit routine. It may be used after the necessary edit operations have been performed. No further display is made of the record read-write operations. Appropriate operator messages are delivered.

3.1.6 PIPPER Key.

This key controls the display of the record arrow which is displayed in the left hand margin of the display screen. The arrow is stepped one record at a time by pressing the key. The arrow is used to indicate which record of those displayed is to be edited.

3.1.7 INSERT Key.

This key signals the main routine to insert a record typed on the bottom line of the display one record ahead of the record indicator arrow. The edit buffer will maintain itself at 15 records by writing the first record of the new buffer on the output tape and repositioning the buffer. The new buffer will be immediately displayed, and further edit operations may be made.

3.1.8 ERASE Key.

This key signals the routine to erase the record indicated by the record indicator arrow. The buffer is processed and immediately redisplayed.

3.1.9 EDIT Key.

This key signals the routine to replace the indicated record with a record typed in on the bottom line of the display. This would seem to be the normal usage of the routine for on-line corrections. The new contents of the edit buffer are immediately displayed.

3.1.10 BACKSPACE Key.

This key signals the routine to backspace the input and output tapes one record each time the key is pressed. It is useful only after the START EDIT key has been pressed and before the READ button is pressed. The READ key causes the input tape to be read for 15 records which will be placed into the edit buffer while the output tape remains stationary until the records have been processed and the WRITE kay has been pressed. The number of records written may be different than the number read into the edit buffer, depending on what operations were performed which would destroy sequencing. Appropriate safeguards are included in the routine, along with operator messages to prevent errors.

3.1.11 SLOWDOWN and STOP keys.

These keys may be used at will during edit operations to either slow down the presentation for readability or to stop it completely.

3.1.12 Miscellaneous Keyboard Information.

All keys for this routine are selectively disabled during normal operations of the satellite station if the MASTER EDIT key has not been pressed. Pressing any key under these conditions merely lights the ERROR key light. Press the ERROR key to reset the light. Error indications will also be given during the edit routine operations if the operator attempts to perform illegal functions. In this case, merely press the ERROR key and repeat correctly. The input tape will be unharmed by any mistakes in editing by the operator; so if necessary, merely begin again.

3.2 Normal Usage.

Use the following control statement:

EDIT, A, B, C.

The arguments are:

A - the tape unit number of the input tape

B - the tape unit number of the output tape

C - mode indicator (0 for normal completion when three blank records have been encountered, or 1 for disabling the blank counter). The final argument need not be used if the input tape ends in three blanks.

3.2.1 Use with Automatic Completion on Three Blanks.

Start the routine without the third argument. The input records will be simultaneously displayed and written on the output tape. Use the SLOWDOWN key as necessary. When the records to be corrected are displayed, press the Start Edit key. The speed of the display has permitted free usage of operator messages which would not be possible with an electric typewriter. When the start message has been delivered, the Read kay should be pressed. Fifteen records will be displayed from the input tape. These may be processed as desired or immediately written on the output tape by pressing the WRITE key. If the operator has allowed the tape display to proceed too far, the backspace key may be used at will until it is disabled by the READ key. To permit some slowness in operator reaction, the tapes are automatically backspaced 10 records before the start message is delivered.

After performing the necessary edit operations, the buffer may be written by pressing the WRITE key. The operator may then press either the READ, START EDIT, or COMPLETE EDIT keys. Appropriate messages are delivered to guide the operator at all times.

3.2.2. Use without Automatic Completion on Three Blanks.

Start the routine using the third argument. Operation is virtually the same, except that automatic completion will be made when an end of file mark is encountered. This mode of operation

has been included to permit edit operations of files which have been built using service routine FILEIN. (See Appendix VIII-A)

Regardless of the value of the third argument, the routine automatically completes itself whenever an End of File mark is encountered on the input tape.

APPENDIX III - B

EDIT ROUTINE - CONTROL

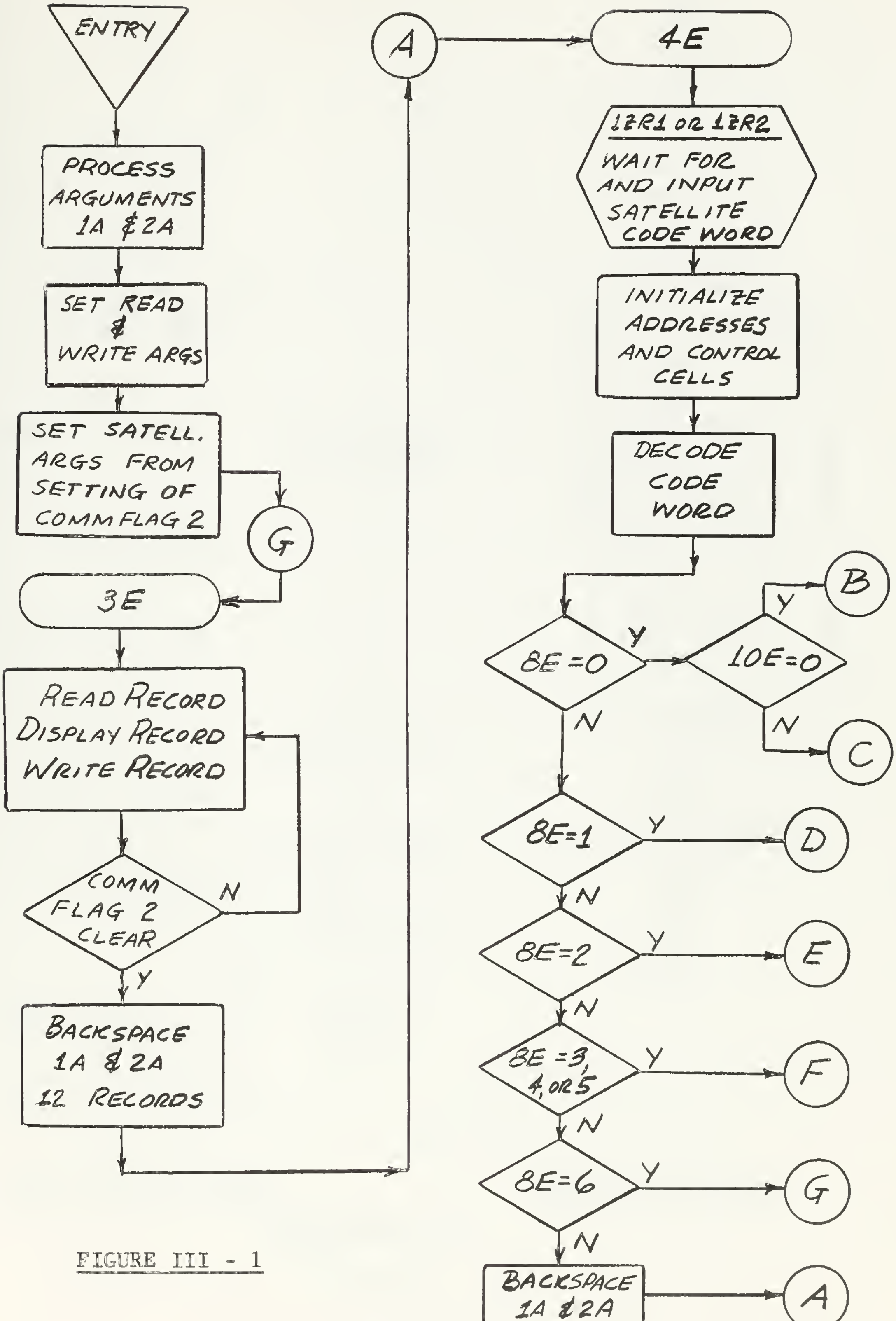
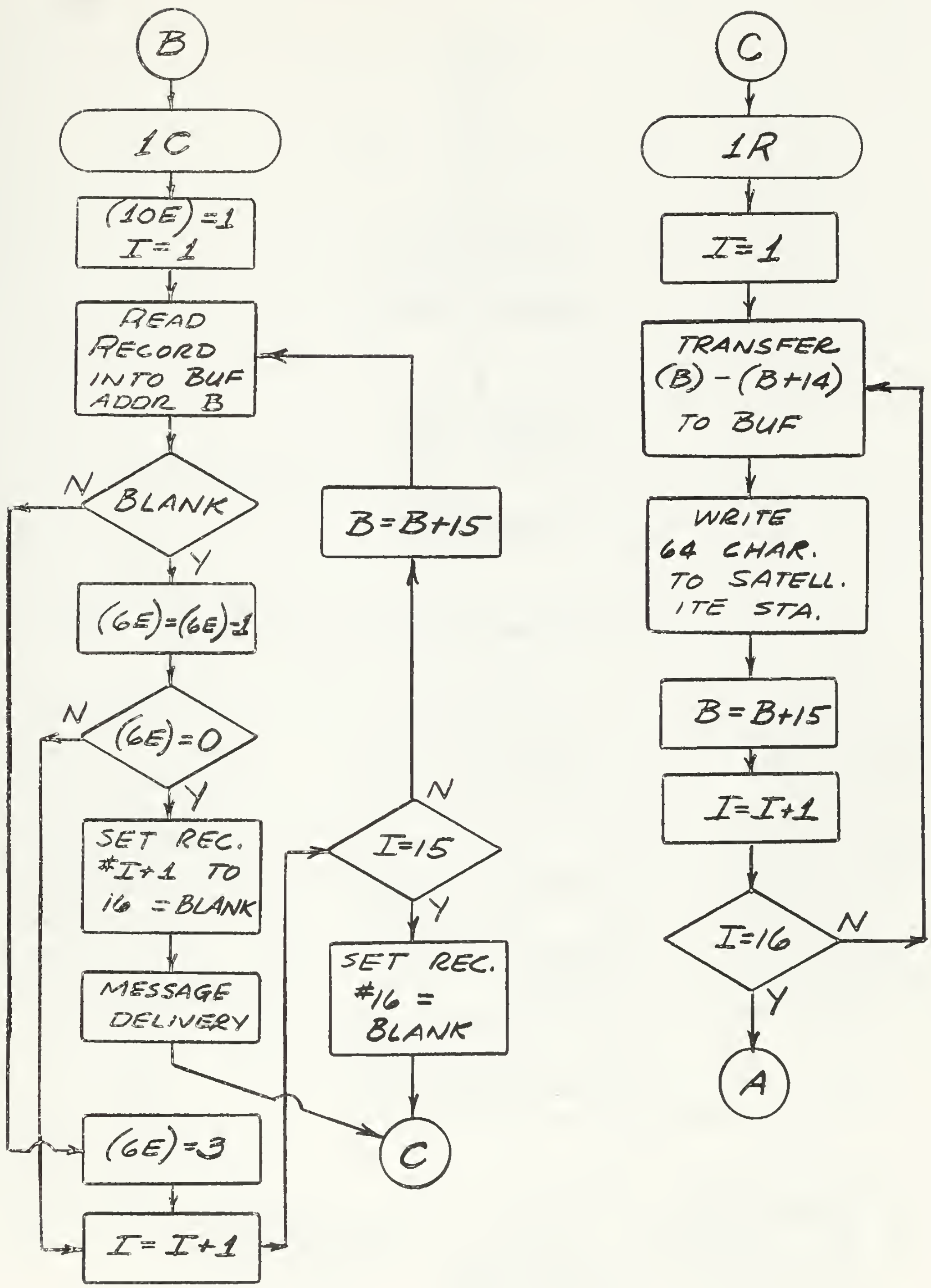


FIGURE III - 1

EDIT ROUTINE



READ INTO EDIT BUFFER

DISPLAY EDIT BUFFER

FIGURE III - 2

APPENDIX III - B

EDIT ROUTINE

PERFORM EDIT OPERATIONS

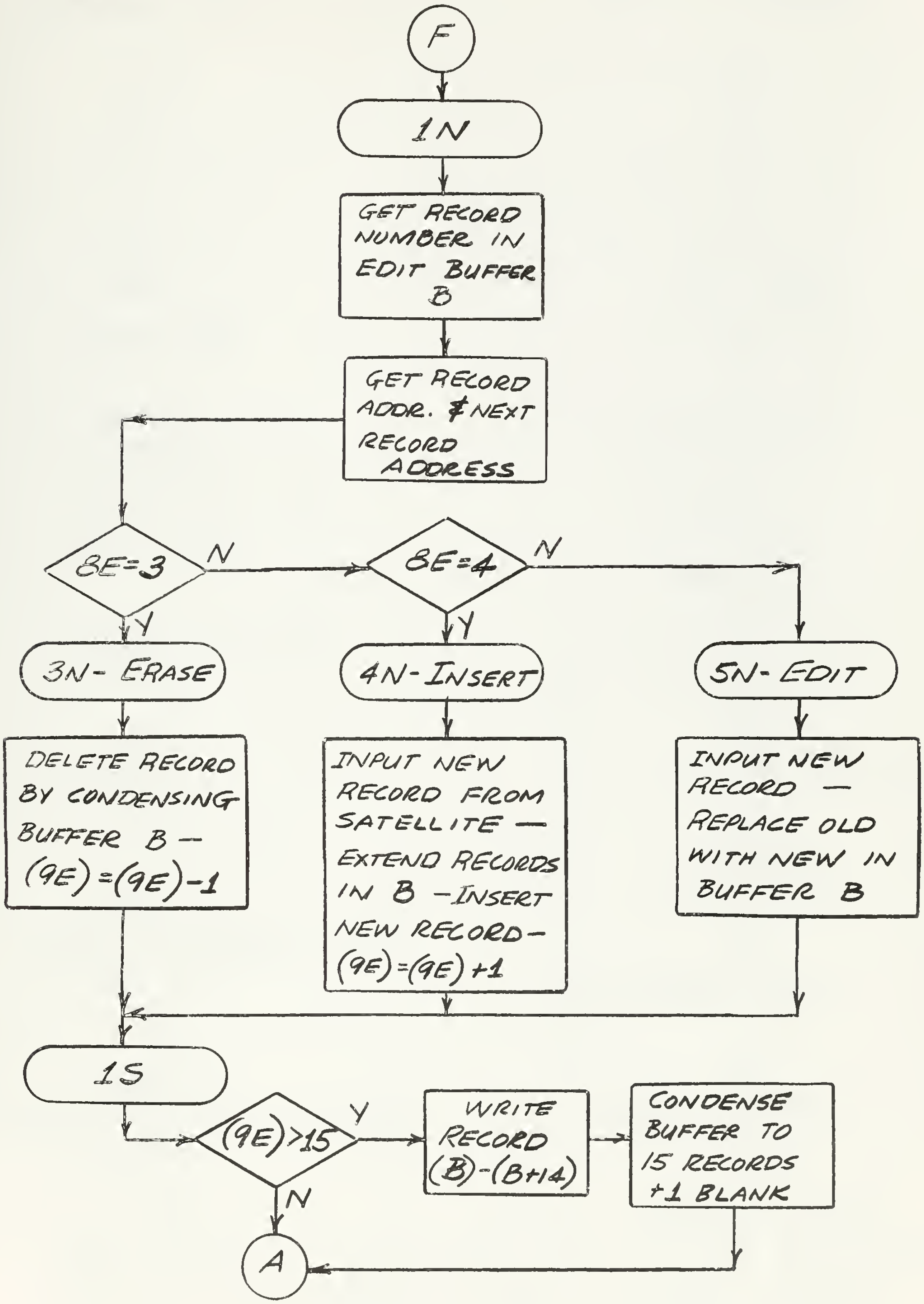
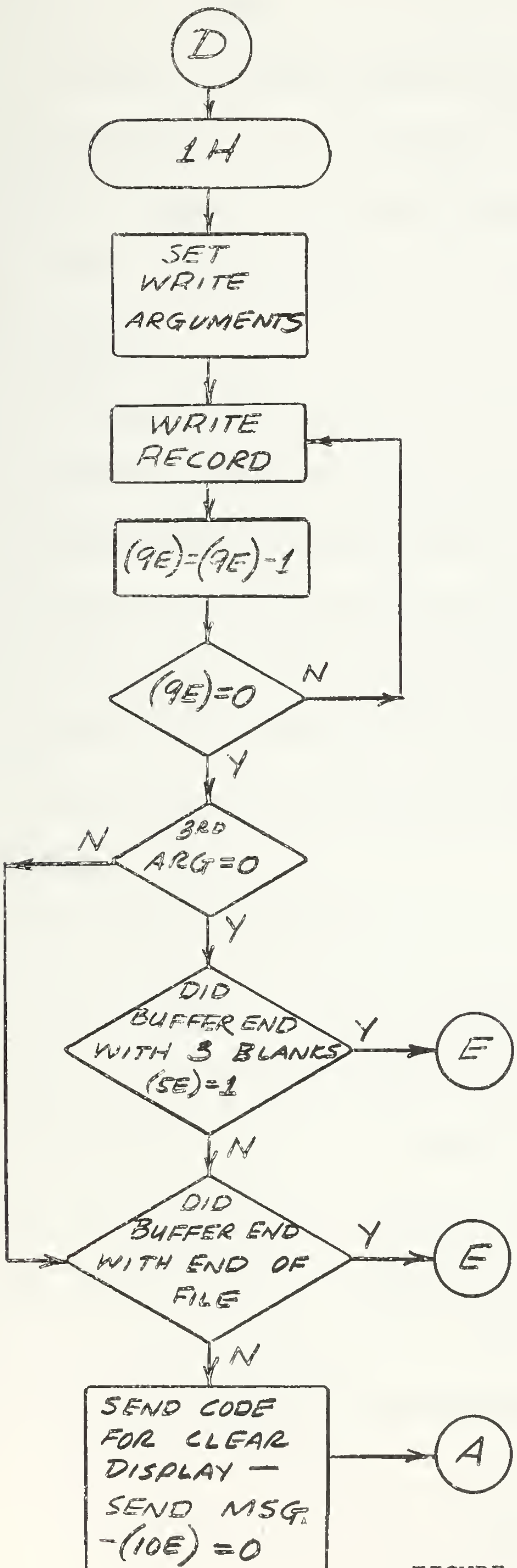


FIGURE III - 3

APPENDIX III - B

EDIT ROUTINE

WRITE EDIT BUFFER



COMPLETE EDIT ROUTINE

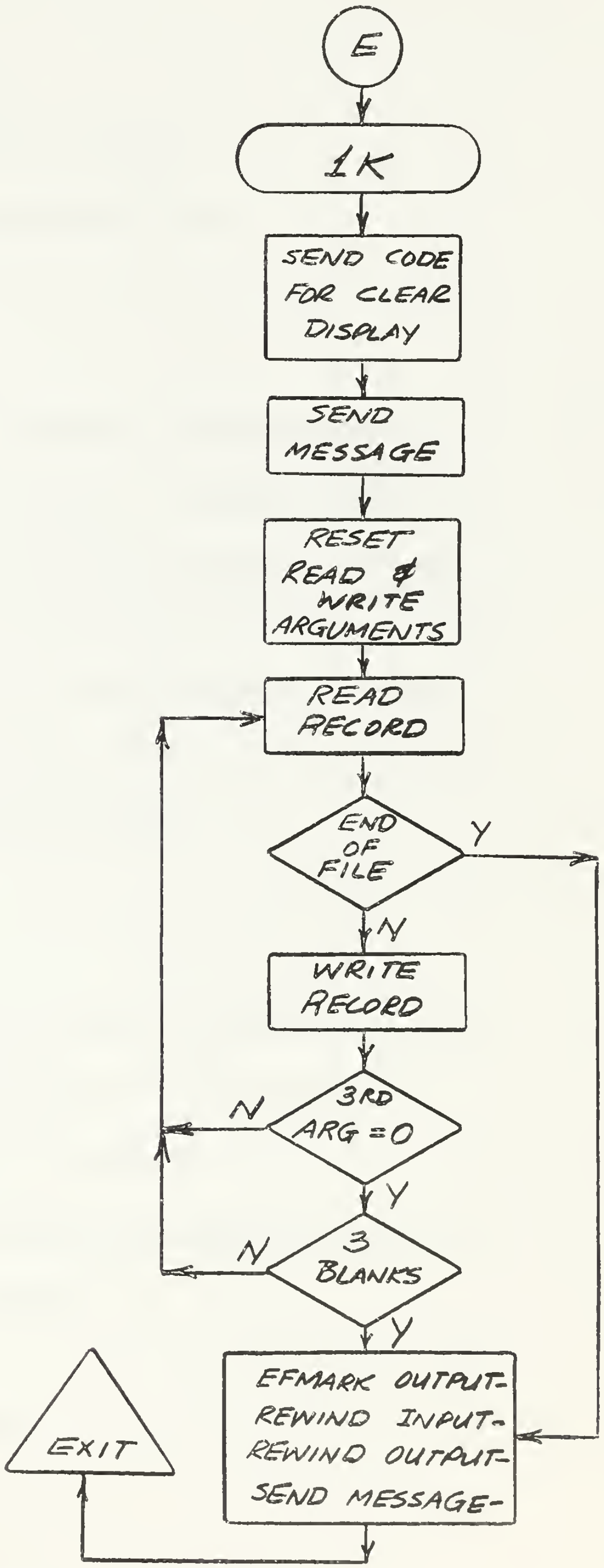


FIGURE III - 4

APPENDIX IV - A

1. Identification

Title: CHANGE

Category: Problem Analysis Aid

Programmers: A. J. Perrella and G. H. Leach

Organization: U. S. Naval Postgraduate School

Date: February 1964

2. Purpose

This subroutine is designed to allow users to make changes to specified variables during the course of a running FORTRAN program by entering these changes from the typewriter. This will allow human decisions concerning the running program to be made immediately, rather than having to recompile with new values for certain parameters. It may also be used to eliminate the necessity for data cards in a general program.

3. Usage

3.1 Normal Operation.

3.1.1 CHANGE provides for entry of floating point (either decimal or exponential format), fixed point, or octal numbers or BCD characters into the designated variables.

3.1.2 In order to illustrate the usage of CHANGE, assume that a program uses the following variables:

X - a floating point array

P - a floating point variable

R(I) - the I th cell in the R array

MM - a fixed point array

NUM - a fixed point variable

OCT - an octal variable

NAME - a BCD tag constant cell

If it is desired to change all these quantitites, then CHANGE may be called as follows:

CALL CHANGE (X,P,R(I), MM, NUM, OCT, NAME)

3.1.3 When the program executes the subroutine, the computer will type BEGIN. The new value for one of the arguments may then be typed, and when complete, it is entered by hitting the carriage return. When using the DD 65 keyboard in satellite mode, press the output button instead of the carriage return. If the format of the change is correct, the computer will enter the new constant into the specified argument location and type back OK. However, if there is any error in the proposed change, the computer will type back ERROR. The number may then be retyped correctly. Additional changes may be made in the same fashion until complete. When all desired changes have been made, typing END and entering it will allow the program to continue. The computer will then type END and exit the routine.

3.1.4 In order to correctly enter a change into the above variables, a transformation of the program name for a variable must

be made to that of the subroutine format. All arguments are referenced as An, where n is the number of the argument in the calling sequence. That is, A2 would be equal to the program variable P, A4(7) would be equal to the program variable MM(7), and so forth.

3.1.5 In illustration, these statements would cause the following changes to be entered:

A1(3) = 5.0 - will set X(3) to -5.0 in floating point
A2 = 1.6E1 - will set P to 16.0 in floating point
A3= 1E-2 - will set R(I) to .01 in floating point
A4(5) = 19 - will set MM(5) to 19 in fixed point
A5 = 100 - will set NUM to 100 in fixed point
A6 = 777B - will set OCT to 777 in octal
A7 = LABELA - will store the BCD characters for LABEL into NAME, left justified, and filled out with spaces

3.1.6 Standard FORTRAN format is used for defining the numbers entered and BCD characters are defined by typing an A after the last desired BCD letter. For BCD inputs, a maximum of eight characters may be entered into a cell.

3.2 Limitations.

3.2.1 CHANGE contains no logic to check on whether a given variable is fixed or floating point, or octal, or BCD within the calling program. Therefore, the user will not get an error

print if he attempts to enter the wrong mode of number into an argument location. The number will be entered as typed, regardless of the mode of the variable within the calling program. Should a mistake of this sort occur before typing END, simply re-enter the number in the correct format.

3.2.2 In order for the user to perform the translation of variable to the appropriate An without confusion, it is suggested that he program a separate PRINT statement prior to a call for CHANGE. For example:

```
PRINT 100
100 FORMAT (//19HVARIABLES ARE X,Y,Z)
CALL CHANGE (X,Y,Z)
```

3.2.3 The maximum number of arguments allowed is nine. Less may be called for if desired. If reference is made to an argument beyond the number called, an error print will result. This subroutine may be called as often as desired during a program, with the same or different arguments as necessary.

CAUTION: Due to an error in the FORTRAN 60 compiler, if the first call for a subroutine uses less arguments then a subsequent call, there is a possibility of generating an error in compilation. This situation may be circumvented by using all nine arguments in the first call, filling in with dummy arguments if necessary. Then

succeeding calls may use any number of arguments. CALL CHANGE (X, Y, Z,,,,,,) used as the first call will avoid all problems. This caution applies to all subroutines with a variable number of arguments. If the FORTRAN 60 compiler is corrected, this caution may be ignored.

3.2.4 Although this subroutine will work with the console typewriter as the control medium, it has greater utility when used in conjunction with the satellite system and the DD 65. Under this control, the program output and on-line graphs may be observed, and changes made to appropriate parameters based on these observations.

3.2.5 This subroutine should not be incorporated into a program that is to run unattended, since it awaits a typewriter response before continuing.

APPENDIX IV - B

SUBROUTINE CHANGE - CHART 1

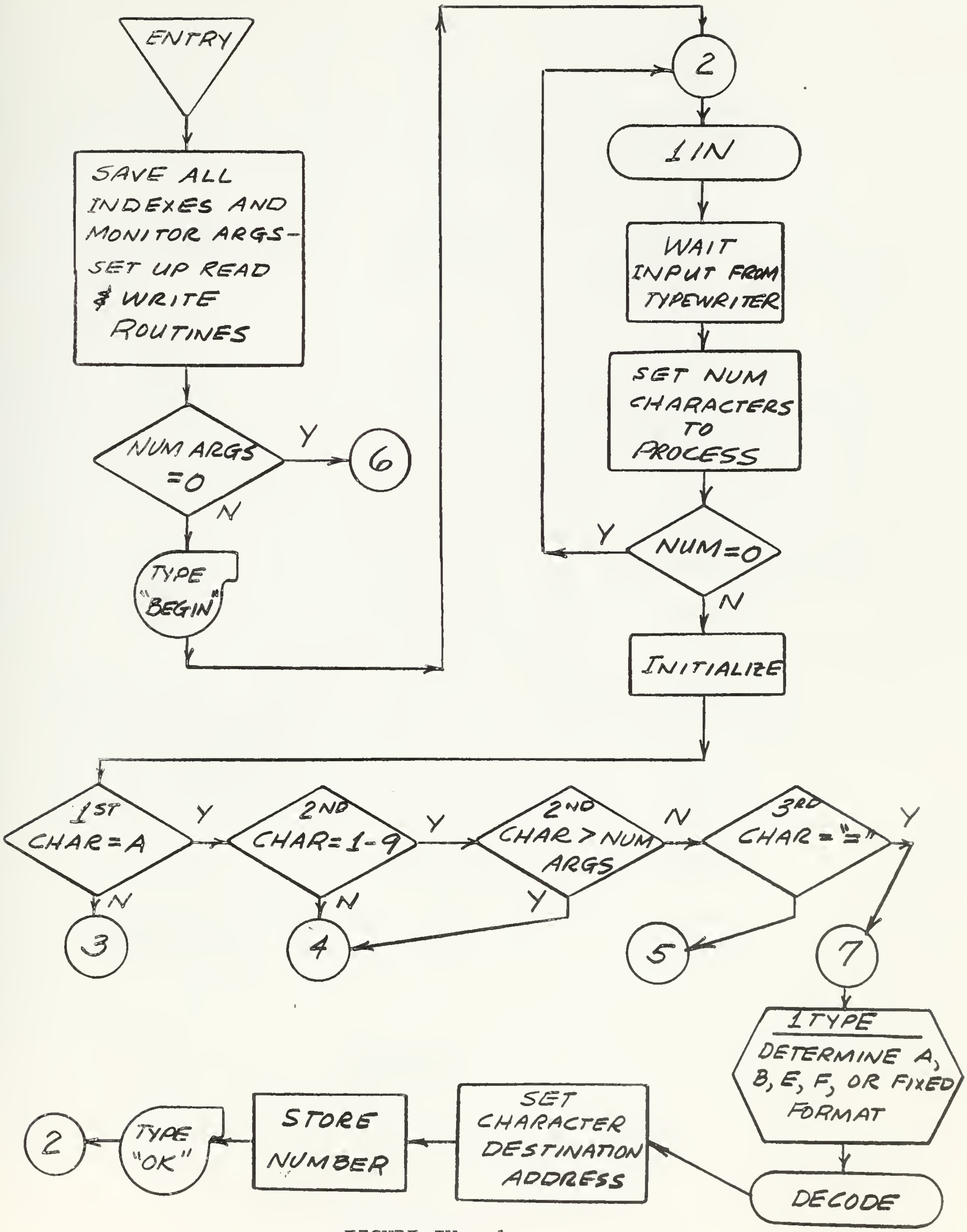


FIGURE IV - 1

SUBROUTINE CHANGE - CHART 2

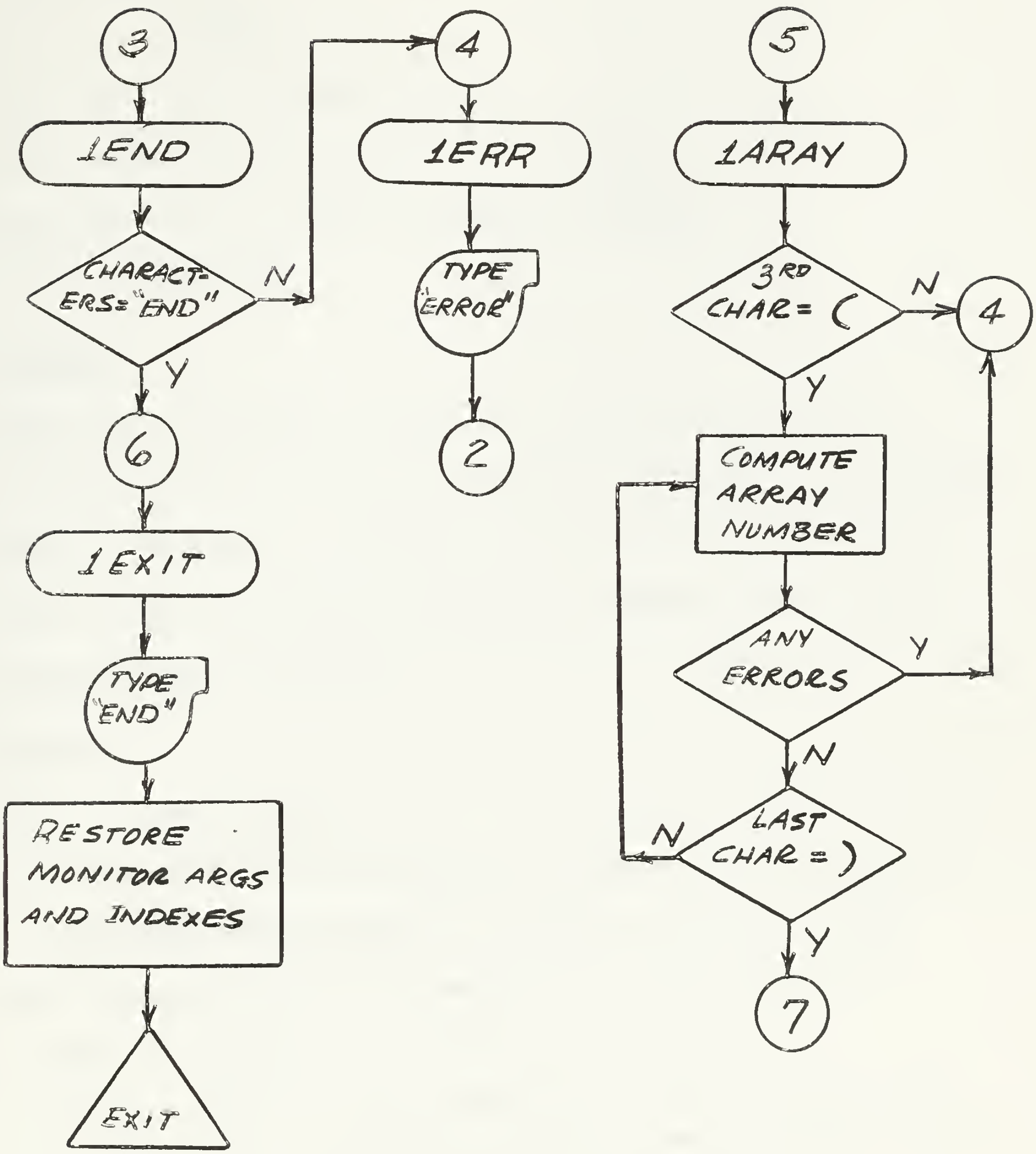


FIGURE IV - 2

APPENDIX V - A

1. Identification

Title: SATGRAF

Category: Problem Analysis Aid

Programmers: A. J. Perrella and G. H. Leach

Organization: U. S. Naval Postgraduate School

Date: March 1964

2. Purpose

This routine enables 1604 satellite system users to observe graphs displayed on the DD 65 Display Unit located in the Digital Control Laboratory, satellite station 2. It enables users to view graphical results, on-line with computation for instant analysis and/or decision.

3. Usage

3.1 Calling Sequence.

The subroutine is called with the following calling statement:

CALL SATGRAF (MODE,NUMPTS, X, Y,LXAX, LYAX, XSCL, YSCL)

The arguments have the following meaning and limitations:

MODE = 0, 1, or 2

- 0 - The graph is to be automatically scaled and positioned.
- 1 - The graph is a follow-on graph to one previously plotted using the same scale and axis positioning.
- 2 - The graph is to be force-scaled and positioned in accordance with following arguments.

NUMPTS is the number of points to be plotted. There is no

upper limit to the number of points, but if NUMPTS is less than 2, an error print out will occur.

X and Y are the names of the arrays to be plotted.

LXAX is the position of the X axis. It must be a fixed point number between 0 and 6. Zero corresponds to the bottom of the display grid, 3 the center, and 6 the top.

LYAX has the same function and limits as LXAX. Zero corresponds to the far left of the display grid, 3 the center, and 6 the far right. If either LXAX or LYAX are not within the specified range of fix point numbers, the graph will be autoscaled.

XSCL is the desired X grid scaling. It must be a floating point number in multiples of 1.0, 2.0, or 5.0.

YSCL is the desired Y grid scaling. It has the same limitations as XSCL. If either XSCL or YSCL are not multiples of 1.0, 2.0, or 5.0, they will be forced to the next higher increment.

3.2 Normal Operation.

3.2.1 In order to utilize SATGRAF, the Digital Control Laboratory (Room 501) CDC 160 and DD 65 must be turned on and the 160 Executive program must be running.

3.2.2 Control of the 1604 by the satellite computer is not essential, but for maximum usefulness, control should reside with the satellite station.

3.2.3 The DD 65 Line Print key must be pressed and lighted or SATGRAF will exit with an operator message, SATELLITE NOT AVAILABLE.

3.2.4 The graph will be plotted on the left tube of the DD 65 display unit, and any associated BCD message output will be displayed on the right tube. These displays are independent. Either may be changed without affecting the other. The graph will remain until a new graph is plotted or the CLEAR key is pressed.

3.2.5 After each graph is plotted, a message on the right tube will appear saying SATELLITE GRAPH PLOTTED. There is a built-in 30 second delay for observation, after which the 1604 will proceed with the program. If the user desires to continue sooner, press the COMM FLAG key (after lighting the Master Signal key). This will cause the 1604 to exit the delay loop immediately. If the user desires to study the graph for a longer period, he may press the Stop key, which will halt any further action at the display and the 1604 until the key is repressed.

3.2.6 Should the first or succeeding follow-on graphs fill the memory of the display unit, further plotting will cease and a message will appear, DD 65 FULL, on the left screen. If no action is taken within 30 seconds, the 1604 will exit the routine with the print out, DD 65 MEMORY FULL, and any succeeding follow-on graphs will similarly exit. A new graph with Mode 0 or 2,

will plot normally. If the operator presses the COMM FLAG key within 30 seconds after the DD 65 FULL message is displayed, the previously plotted points will be cleared and the remaining points will be displayed.

3.2.7 The grid scaling, whether automatic or forced, will appear in the lower left portion of the left tube for reference. Should an error be made in scaling, a message SCALING ERROR will be displayed and the routine will exit.

3.2.8 All eight arguments are necessary in the calling statement only when using forced scaling (Mode = 2). In other cases, only the first four arguments need be used.

CAUTION: Due to an error in the Fortran 60 compiler, if the first call for a subroutine uses fewer arguments than a following call, there is a possibility of generating an error in compilation. This situation may be circumvented by using all eight arguments in the first call, with the last four arguments as dummies if necessary. Then, succeeding calls may use any number of arguments. CALL SATGRAF (0, 900, X, Y,,,,) used as the first call will avoid all problems. This caution applies to all subroutines with a variable number of arguments. If the Fortran 60 compiler is subsequently corrected, this caution may be ignored.

3.3 System Usage.

3.3.1 The subroutine SATGRAF may be safely imbedded in any program that will be compiled with the FORTSHARE library. If the satellite station is not in operation, the program will exit the routine with an appropriate print-out on the output medium. Calling SATGRAF on a library other than FORTSHARE will generate compilation errors.

3.3.2 The subroutine is designed to proceed normally, provided the satellite system is in operation, without the necessity for operator intervention. The COMM FLAG key directs exit from the standard 30 second delay loop after a normal plot, and thus permits the operator to speed operations.

3.3.3 In order to take full advantage of the subroutine SATGRAF, it is advisable to also use subroutine CHANGE, whereby critical parameters may be changed during the course of the program and the effects of the changes observed graphically. This procedure should enable much quicker optimization of a solution than having to examine hard copy graphs off-line.

3.3.4 As many graphs as desired may be plotted on the same axes as follow-on graphs (Mode = 1); however, due to limited DD 65 memory capacity, it is probable that no more than 10 may be plotted without overflow.

4. Method

4.1 Control System.

4.1.1 The SATGRAF routine uses the satellite control system built into the FORTSHARE resident routine, in conjunction with the CDC 160 computer and DD 65 Display Unit. The operational aspects of this system are discussed in Appendix I and II of this paper.

4.2 Equipment Constraints.

4.2.1 Graphs are drawn using the vector mode of the DD 65 Display Unit. Since vectors may only be drawn in increments of 45 degrees and the smallest length vector is 1/16th inch, the graphs are of necessity somewhat rough. However, using the algorithm developed with Hogg and Glover (16) for best fit between two points, the displayed graph is accurate to within 1/32nd inch maximum deviation. Should modification of the DD 65 Display Unit permit shorter vector length, modification of the SATGRAF routine would allow smoother graphs, although the apparent memory capacity of the unit would be reduced.

4.2.2 When force scaling any of the axis positions, 0 to 6 may be used. When using autoscale, however, only positions 1 to 5 will be selected. Therefore, when using autoscaling, there will be some allowance for a follow-on graph to be

slightly larger than the first. If a graph exceeds the grid boundaries, the part which is off-scale will not be plotted, but if it reenters the grid boundaries, the plot will be continued.

APPENDIX V - B

SUBROUTINE SATGRAF - CHART 1

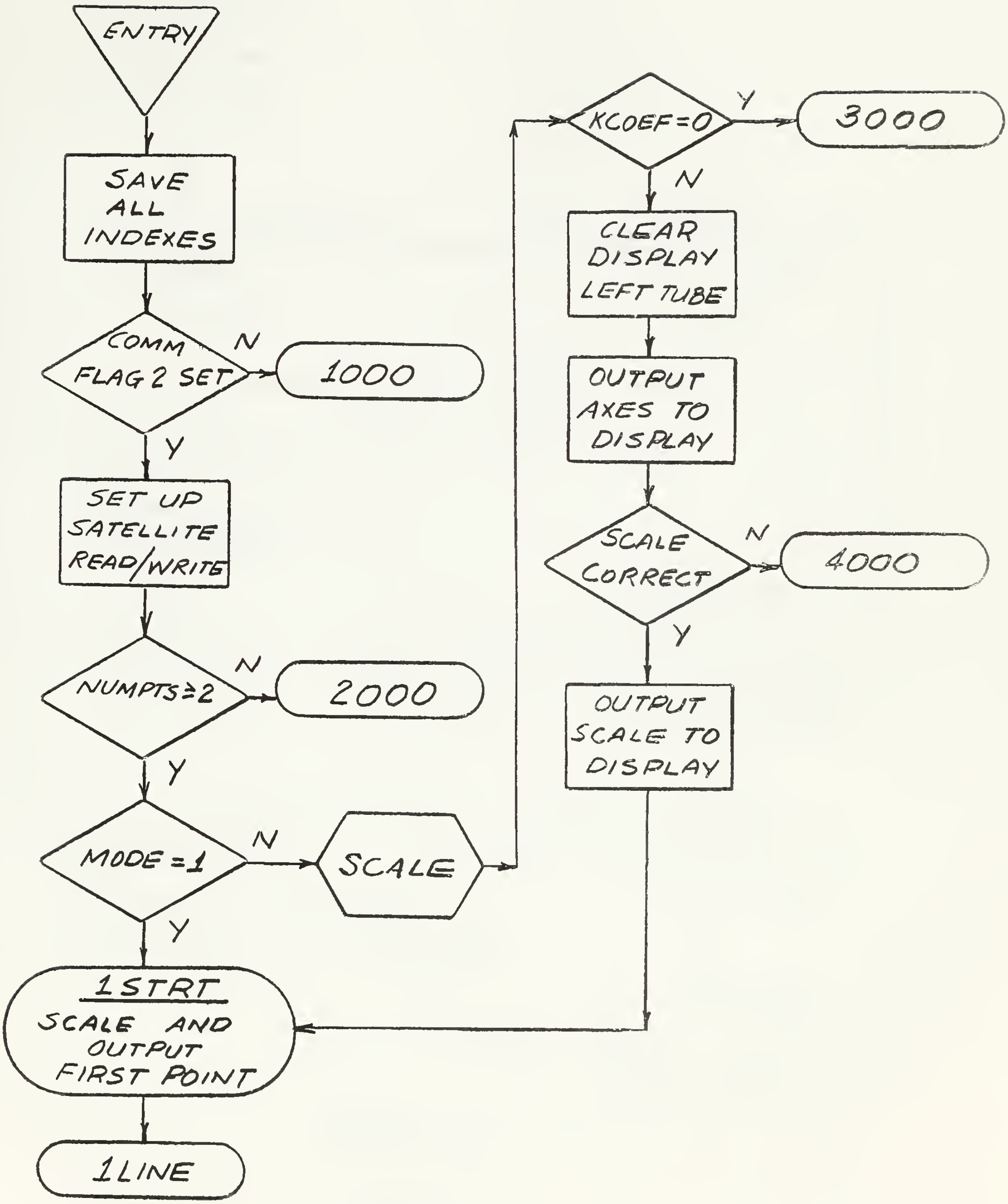


FIGURE V - 1

APPENDIX V - B

SUBROUTINE SATGRAF - CHART 2

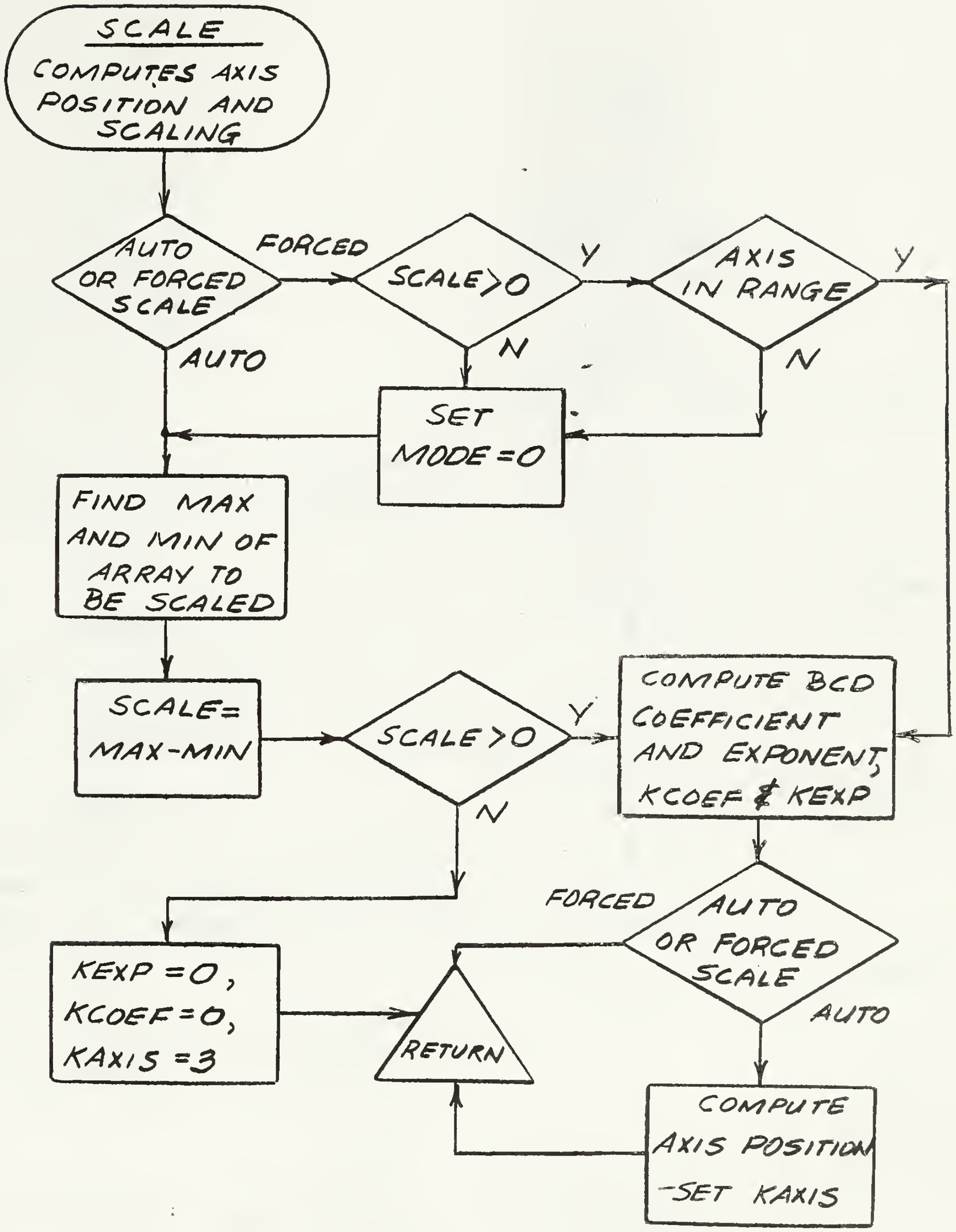


FIGURE V - 2

APPENDIX V - B

SUBROUTINE SATGRAF - CHART 3

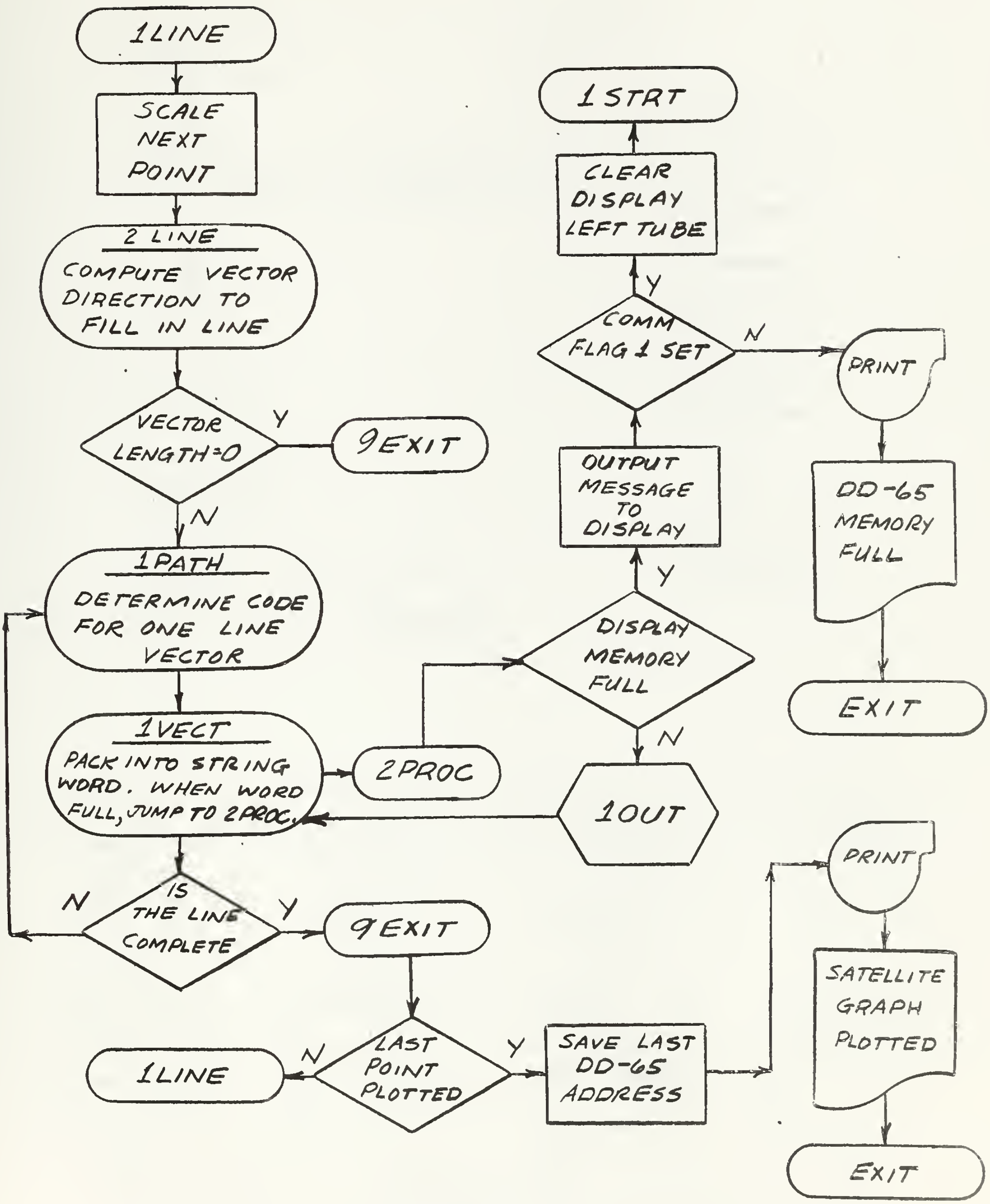


FIGURE V - 3

APPENDIX V - B

SUBROUTINE SATGRAF - CHART 4

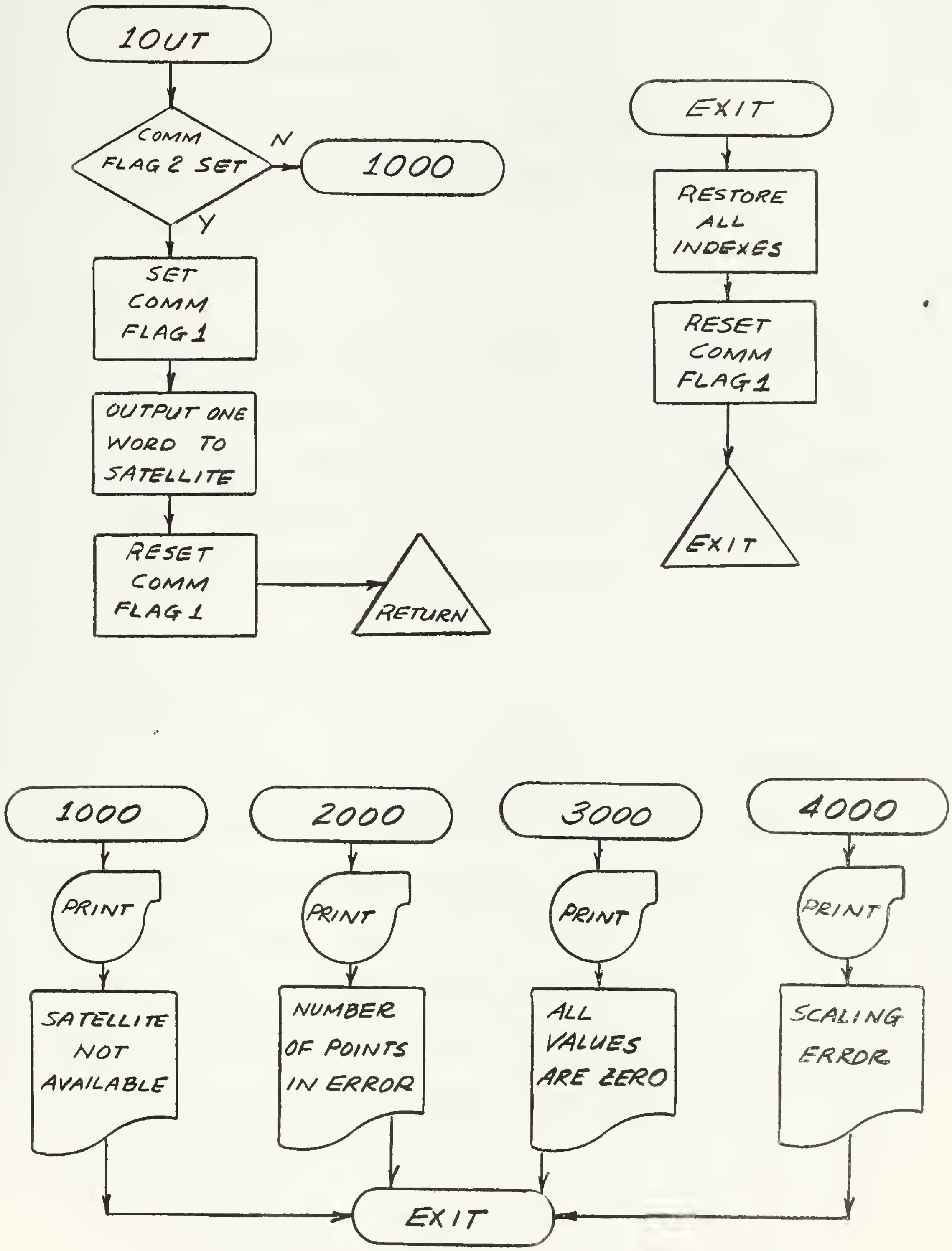


FIGURE V - 4

APPENDIX VI - A

1. Identification

Title: 160 BOOTSTRAP

Category: Satellite Routine

Programmers: G. H. Leach and A. J. Perrella

Organization: U. S. Naval Postgraduate School

Date: February 1964

2. Purpose

This routine enables satellite computer users to call into 160 memory any program currently available in the file of satellite computer programs located on the main computer library FORTSHARE. Appendix VII covers procedures for adding satellite computer programs to the main library.

3. Usage

3.1 Method.

3.1.1 The satellite computer interrupts the main computer, delivers a code word informing the computer which task is to be executed under interrupt lockout, and informs the main console that a library search is in progress. The main computer, by means of the code word, picks up the title of the program desired, searches the library till found, then delivers the program to the satellite computer. The routine may be run at any time that FORTSHARE is mounted, and requires from 3 to 7

seconds to complete. Longer periods of lockout may occur should the main computer program be out searching the library for subroutines during the final stage of compilation. In this case, a delay of up to 20 seconds is possible. When the call is complete, the value indicated in the A register should agree with those shown on the program sheet.

3.1.2 The satellite computer programs are stored in the main library in standard 54 word blocks, the first word of which is the title. This means that the satellite computer memory is loaded in bytes of 4 x 53 words, since each 1604 word is equivalent to four 160 computer words. The details of the 1604 search program may be found in Appendix I.

3.2 Normal Usage.

3.2.1 Load BOOTSTRAP paper tape in the 160 computer at any location. It should be remembered that the called program will be read in in 324_8 word blocks. Should the BOOTSTRAP routine be loaded at too low a position, the called program may write over the routine and stop the computer. For these reasons, it is good practice to load BOOTSTRAP at location 7700 for all programs except RS022.

3.2.2 Satellite Station 2 may reference the main control and satellite control console lights. Satellite control is lighted during library search in the BOOTSTRAP routine. Satellite Station 1 has no lights.

3.2.3 Enter A with the constant corresponding to the program desired. See the current list of programs available. RUN. The program will stop with P = L + 7 and A = 0000. Enter A with the desired initial load address of the called program and RUN. On stop, the A register indicates the terminal load address +1.

3.2.4 Clear and run as desired.

3.2.5 An illegal constant will cause the program to search to the end of the 160 program area on the main computer library. When the program halts, A will equal the initial load address. No memory will have been loaded.

APPENDIX VI - A

SATELLITE COMPUTER PROGRAMS AVAILABLE WITH BOOTSTRAP

TITLE	A CONSTANT	LOAD ADDRESS INIT - TERM	NUM BLOCKS TRANSFERRED	NORMAL A STOP
RSO22	1	7400 - 7776	2	0251
OSAP MODI	2	0000 - 3760	12	4110
OSAS 0000	3	0000 - 5061	15	5304
TEXPAC	4	0000 - 4642	14	4760
FLOAT	5	ANY -	2	L+650
GRAPH PLOT	6	0000 - 3652	12	4110
CARDS TO PAPER TAPE	7	0000 - 0560	2	0650
CARDS TO MAG TAPE	10	0000 - 0300	1	0324
OSAS 1010	11	0000 - 4673	14	4760
OSAS LOADER	12	ANY - L+600	2	L+650
OSAP LOADER	13	ANY - L+117	1	L+324
SATELLITE CONTROL	14	0000 - 2554	7	2714

TABLE VI - 1

APPENDIX VI - B

SATELLITE BOOTSTRAP ROUTINE

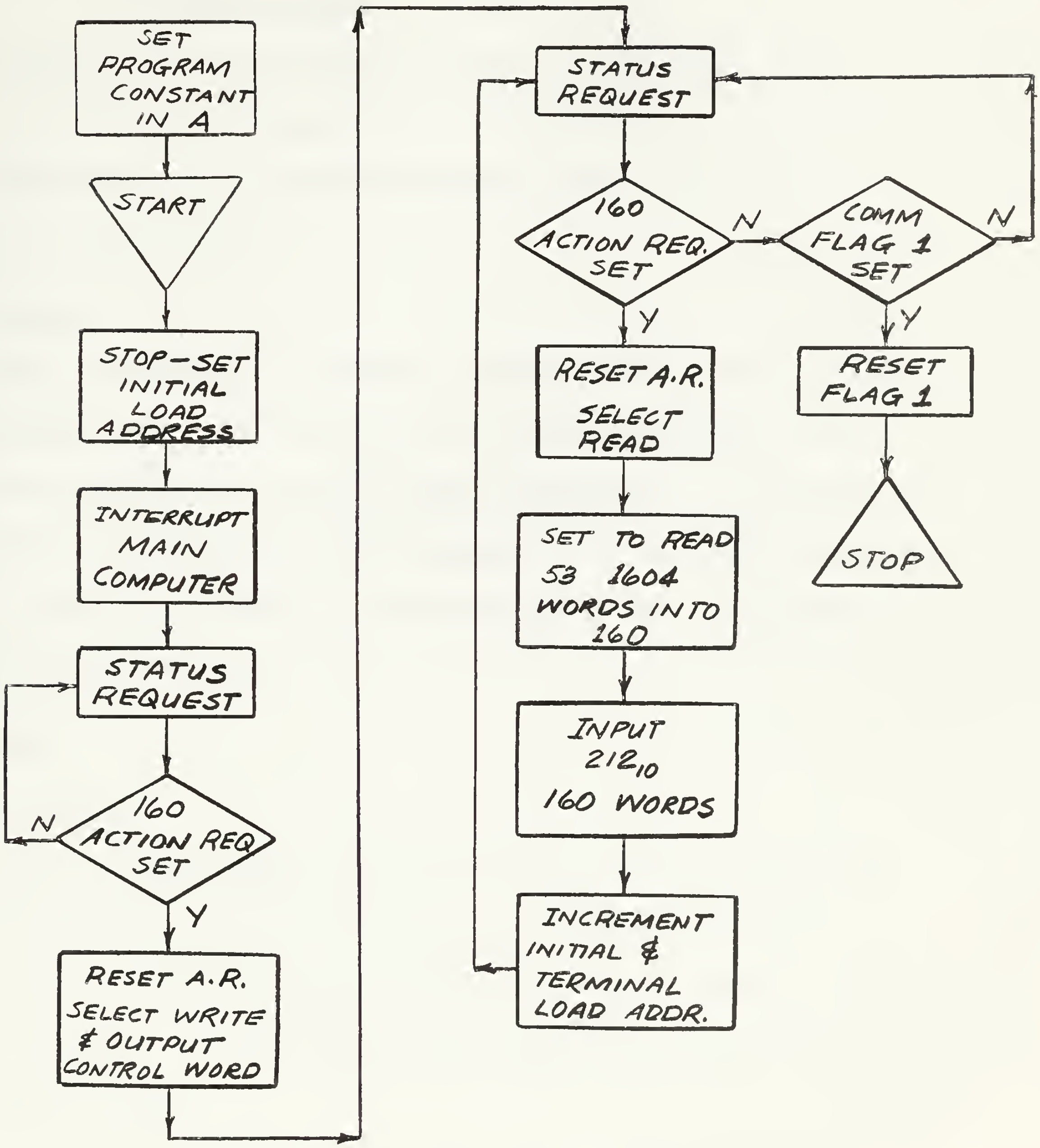


FIGURE VI-1

APPENDIX VII - A

1. Identification

Title: UPDATE

160 UPDATE ROUTINE

Category: Q (Service Routine) and 160, respectively.

Programmers: G. H. Leach and A. J. Perrella

Organization: U. S. Naval Postgraduate School

Date: January 1964

2. Purpose

These routines provide the means for adding satellite computer programs to the callable file of 160 computer programs located on the main computer Fortran library FORTSHARE. The 1604 routine UPDATE is a typewriter callable service routine. The 160 routine is a very short paper tape loaded program which is integrated with the 1604 program.

3. Usage

3.1 Method.

3.1.1 The 160 program does not interrupt the main computer but waits until interrogated by the main computer before transferring its program where it is processed by service routine UPDATE into standard library format binary blocks, which are 54 1604 words in length.

3.1.2 The title of the 160 program is written in the first word of the first block. The first word of each succeeding

block is left blank (all binary zeros). Fifty-three words of the block contain information from the 160 program, and thus 212 160 words must be transferred to make up each block (324_8).

3.1.3 The programmer must figure the number of 324_8 word blocks contained in his 160 program, since this quantity is used as an argument in UPDATE.

3.1.4 When the program is called off the library by BOOTSTRAP (see Appendix VI), the 160 memory is loaded in blocks of 324_8 words.

3.1.5 The routine may be run from either satellite station. (Note - For use with the computer facility 160 computer, Channel 3-4 function switch in tape bank 1 must be in Program Control position).

3.2 Normal Usage.

3.2.1 Load the 160 program to be added to the library. Load paper tape UPDATE at any convenient location above this program.

3.2.2 Call service routine UPDATE from the library of the main computer. Place the 160 computer in RUN at the 160 UPDATE load address with A = initial address of program to be written. The 160 will now wait until called upon by the 1604.

3.2.3 At the main computer console, type:

UPDATE, A, B, TITLE.

A is the number of 324_8 word blocks to be transferred from the 160 to the next higher octal integer.

B is the tape unit on which the program will be written in library format.

TITLE is the name of the 160 program, eight characters maximum, the first of which must be alphabetical. For the sake of library listing continuity, make the last three characters "160".

3.2.4 The 160 program will have been written on the designated tape unit. The final block is a binary "ENDFILE" and the tape will have been backspaced. It is thus immediately ready for additional UPDATE operations and is completely compatible with library service routines LIST and TRANSFER.

3.2.5 The scratch library may now be transferred to the main library in the area immediately following RESIDENT but prior to QUIT160. QUIT160 is a pseudo search tag to prevent search of the entire library should an improper call argument be entered with the BOOTSTRAP routine.

3.2.6 The 160 program UPDATE will stop with the A register equal to the terminal address transferred plus one. See Table VII-1 for Transfer Conversion Constants.

3.2.7 The program name should be added to the list of 160 Library Call program names in FORTSHARE resident for the next library recompilation. See (19), (Resident Listing)

for format. Add the information to the list of available 160 programs (Table VI-1).

APPENDIX VII - A

160 TRANSFER CONVERSION CONSTANTS

Block No. (Dec)	Block No. (Oct)	Dec Add.	Oct Add.
1	1	212	324
2	2	424	650
3	3	636	1174
4	4	848	1520
5	5	1060	2044
6	6	1272	2370
7	7	1484	2714
8	10	1696	3240
9	11	1908	3564
10	12	2120	4110
11	13	2332	4434
12	14	2544	4760
13	15	2756	5304
14	16	2968	5630
15	17	3180	6154
16	20	3392	6500
17	21	3604	7024
18	22	3816	7350
19	23	4028	7674

TABLE VII - I

APPENDIX VII - B

1604 UPDATE ROUTINE

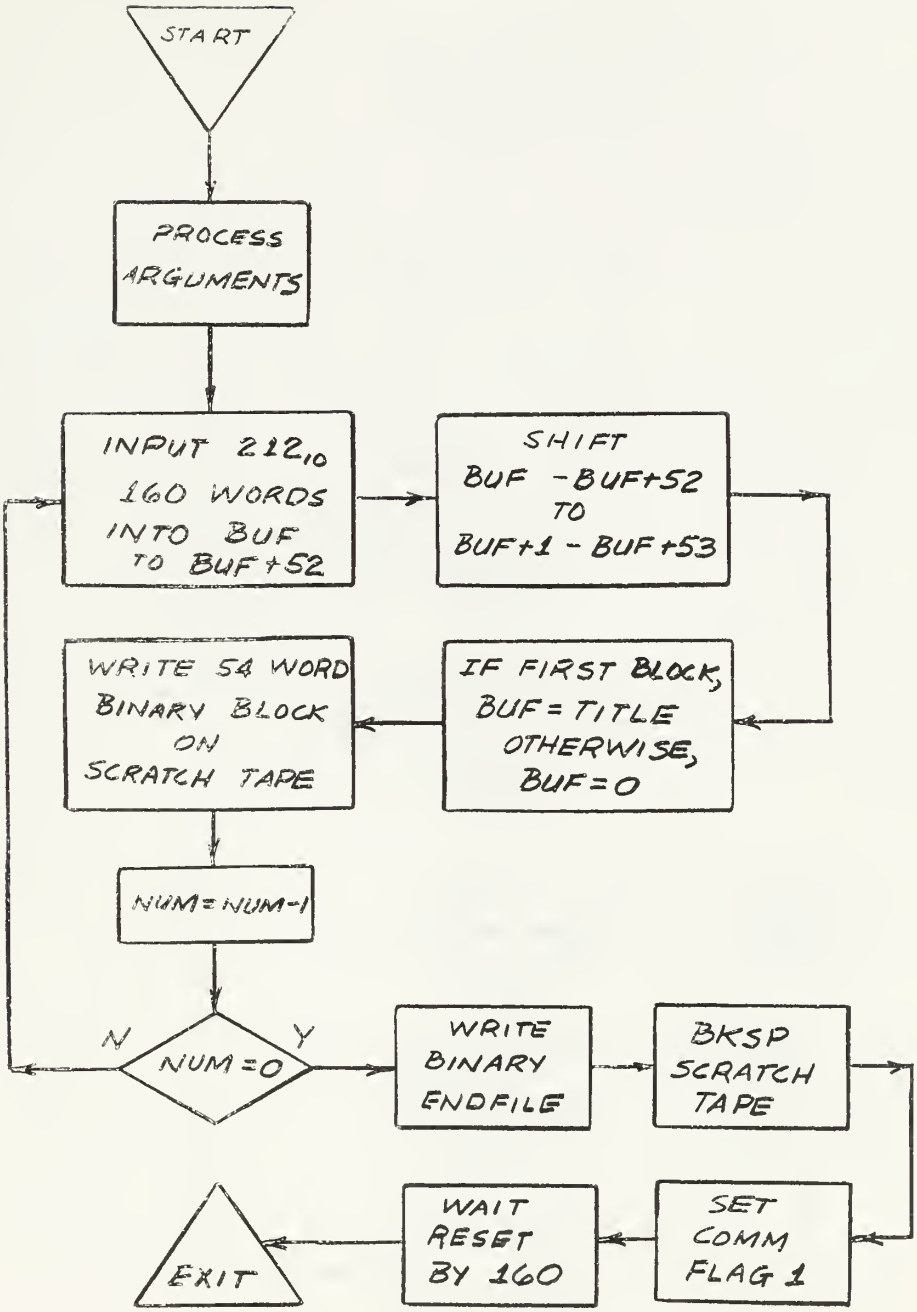


FIGURE VII - 1

160 UPDATE ROUTINE

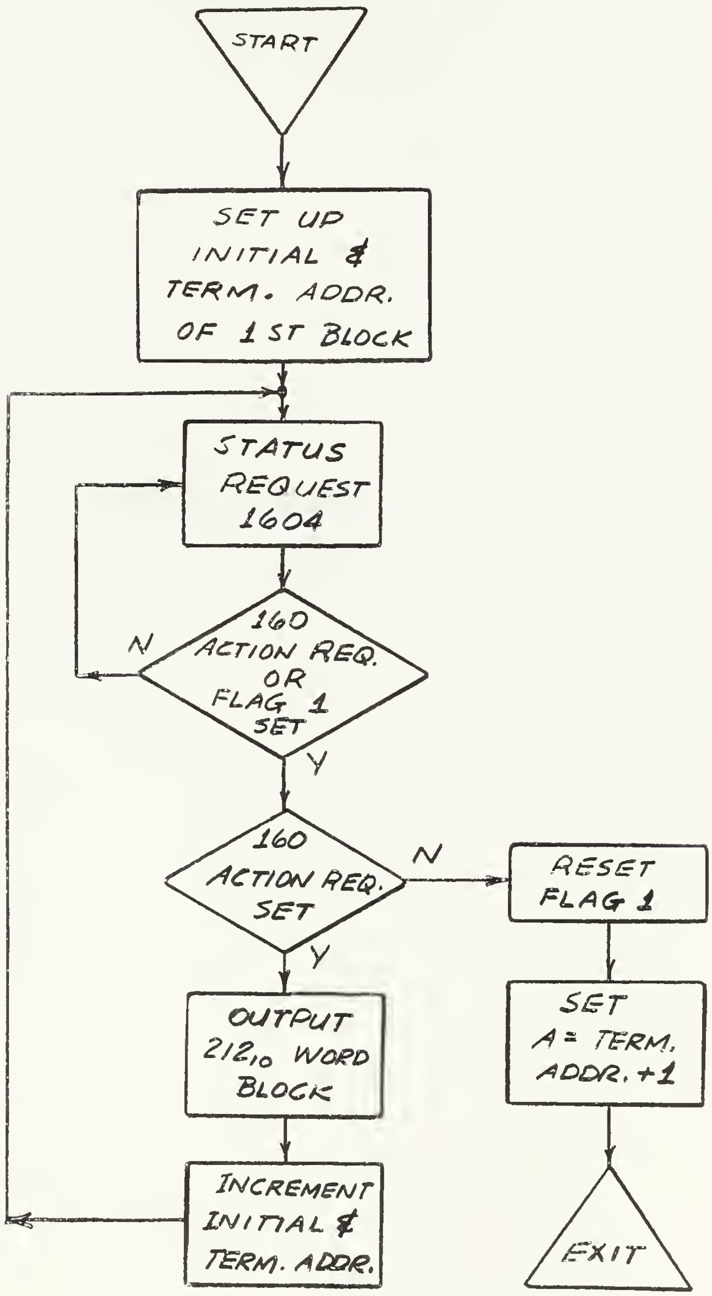


FIGURE VII - 2

APPENDIX VIII

SERVICE ROUTINES FOR SYSTEM

USAGE

APPENDIX VIII-A

1. Identification

Title: FILEIN

Category: Q (Service Routine)

Programmers: G. H. Leach and A. J. Perrella

Organization: U. S. Naval Postgraduate School

Date: April 1964

2. Purpose

This routine is a typewriter controllable service routine for storing programs or file material of 120 character BCD records. The program or file material is retrievable with service routine FILEOUT (see VIII-B).

3. Usage

3.1 Normal Usage.

The routine is typewriter callable from the library and will be called automatically with FORTSHARE when the special call SR is used. Input program or file material should be ended with three blank cards. The program places the records of the input tape on the output take with a special search tag identifier.

3.1.1 Example of usage.

Use the following calling sequence:

FILEIN, A, B, CCC, D.

The arguments are:

A is the input tape unit number

B is the output tape number

CCC is the name of the program or file (a maximum of eight characters, the first of which must be alphabetical)

D is the mode of operation (0 for normal completion when three consecutive blank records are encountered, or 1 for completion on an End of File Mark). This argument is not necessary for use with programs with three blanks at the end, or where only a single program or file is located on the input tape. The option of completion on three blank records has been provided to allow handling more than one program on the input tape. Multiple programs need only be separated by three blank cards.

3.1.2 At the completion of the routine, the output tape is left properly positioned for another FILEIN operation if desired. The input tape will also be left positioned should more than one program or file have been included in the input.

APPENDIX VIII - AA

FILEIN

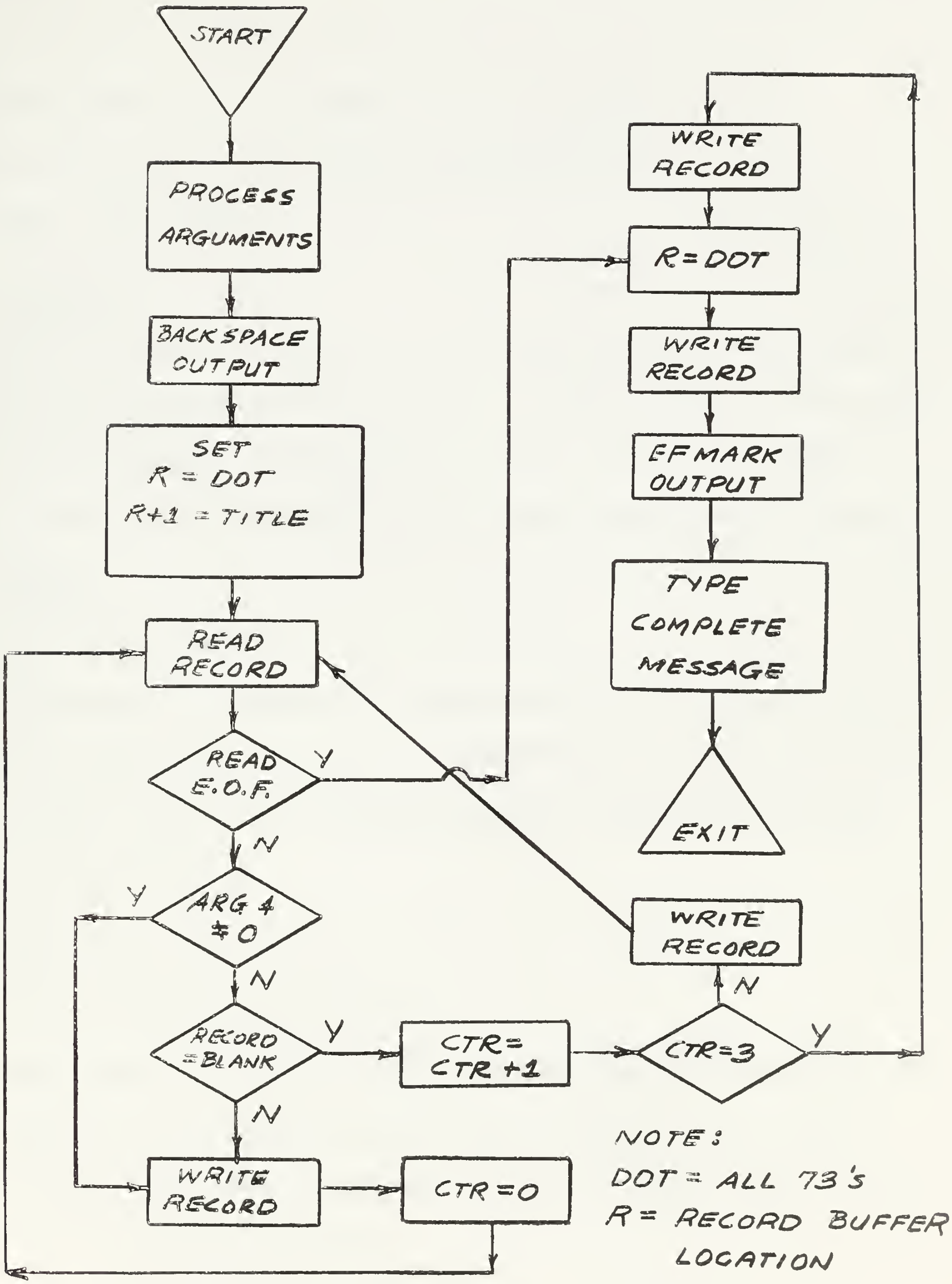


FIGURE VIII - 1

APPENDIX VIII - B

1. Identification

Title: FILEOUT

Category: Q (Service Routine)

Programmers: G. H. Leach and A. J. Perrella

Organization: U. S. Naval Postgraduate School

Date: April 1964

2. Purpose

This routine is a typewriter controllable service routine for retrieving programs or file material of 120 character BCD records which have been placed in the input file tape by the FILEIN routine. Flexibility is provided with several modes of operation.

3. Usage

3.1 Normal Usage.

The routine is typewriter callable from the library and will be called automatically with FORTSHARE when the special call SR is used. The file of programs previously made up using FILEIN is the input tape.

3.2 Modes of Operation.

3.2.1 Positioning Master File.

This mode of operation provides for positioning the input file (which has been assembled with routine FILEIN) at the first record of the desired program for use as an input take for Monitor or compiler operations.

3.2.2.1 Example of Usage.

The following control statement should be typed on the console typewriter:

FILEOUT, A, X, NAME.

where A is the tape number of the input file

X is a dummy argument and must be used

and NAME is the name of the program or file desired.

When the tape is properly positioned, a message will be delivered to the console typewriter. Should the program not be found, an appropriate message will be delivered.

Typing ENDFILE as the last argument in the calling statement above will cause the input tape to be positioned at the extreme end of the master file. At this time, new programs may be added.

3.2.2 Copying a Program to Another Tape.

This mode of operation provides for searching the input tape until the desired program is found. When found, the program is written on the designated output tape.

3.2.2.1 Example of Usage.

The following control statement should be typed:

FILEOUT, A, B, NAME.

where A is the tape unit number of the input

X is a dummy argument and must be used.

The routine completes when an End of File mark is encountered and the input tape will be rewound.

3.2.4 Deleting a Portion of the Master File.

This mode of operation has been provided to permit removal of one program from the master file. This is accomplished by rewriting the master file on a new output tape, with the exception of the desired program to be deleted.

3.2.4.1 Example of Usage.

Type the following control statement:

```
FILEOUT, A, B, NAME, 1.
   where A is the input tape unit number
         B is the output tape unit number
         NAME is the name of the program to be deleted
              from the new master file
         1 is a signal flag argument and must be used
```

When the routine is complete, the output will be left positioned and the input will be rewound. An appropriate message will be delivered to the console typewriter.

APPENDIX VIII - BB

FILEOUT - CHART 1

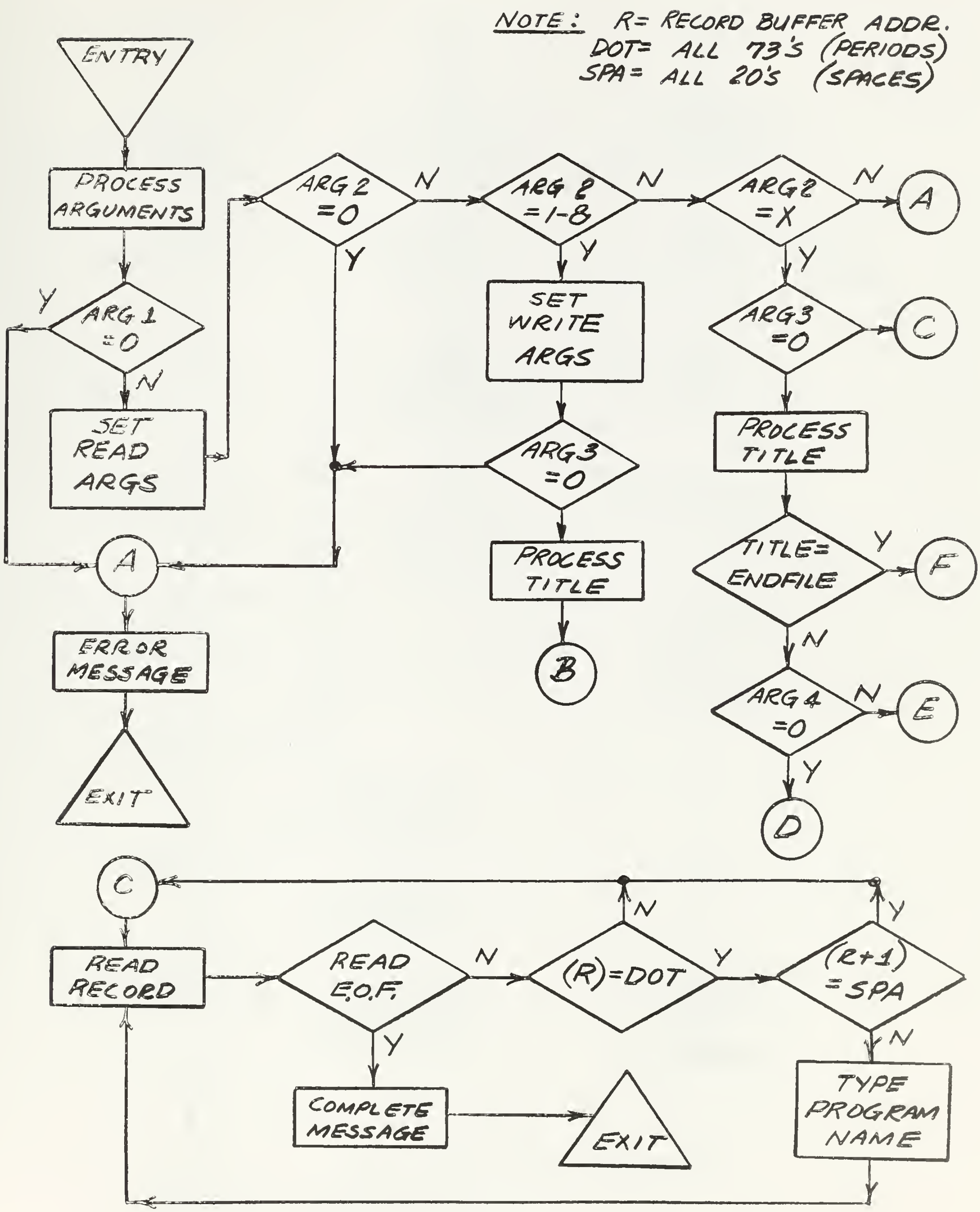


FIGURE VIII - 2

APPENDIX VIII - BB

FILEOUT - CHART 2

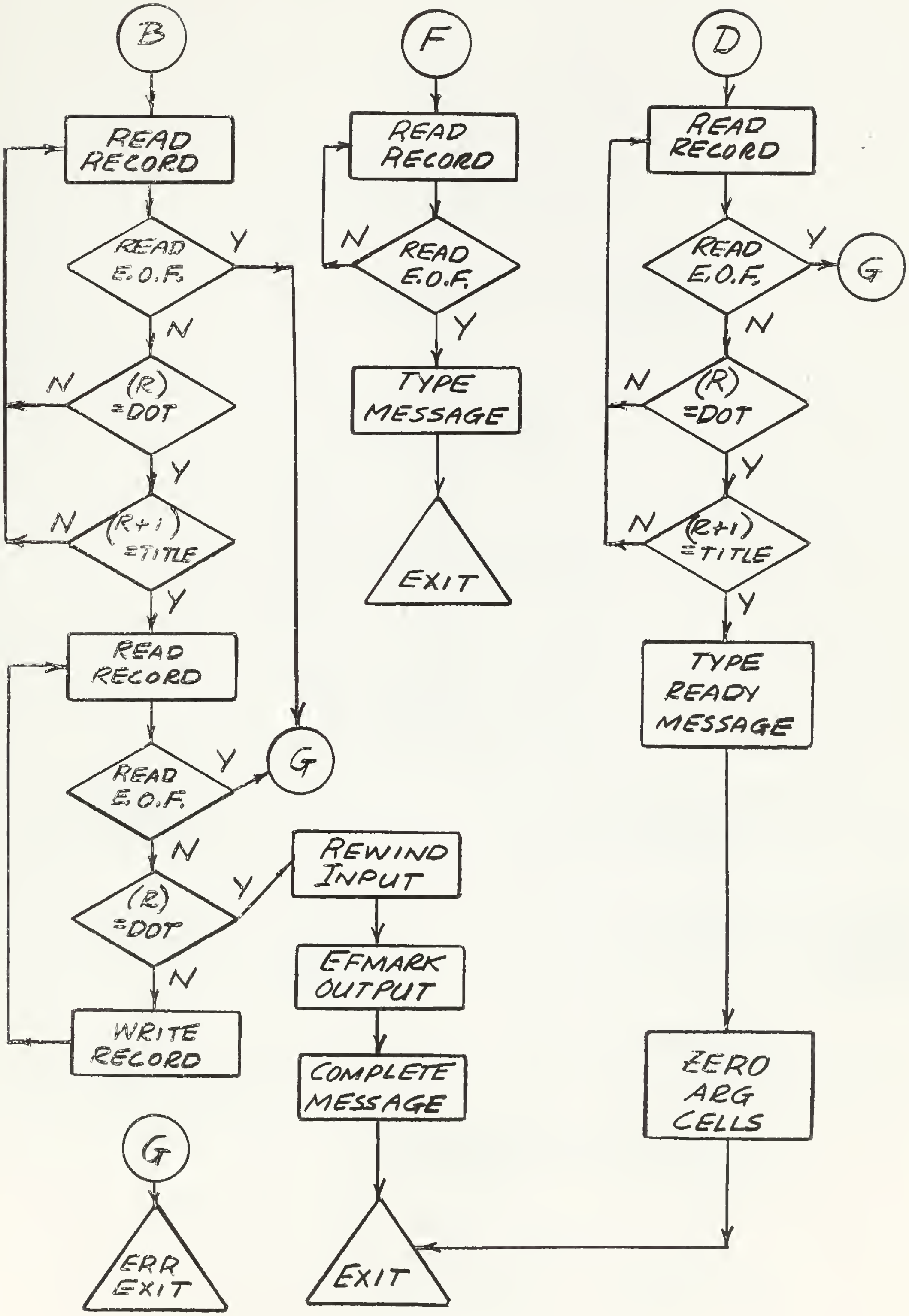


FIGURE VIII - 3

APPENDIX VIII - BB

FILEOUT - CHART 3

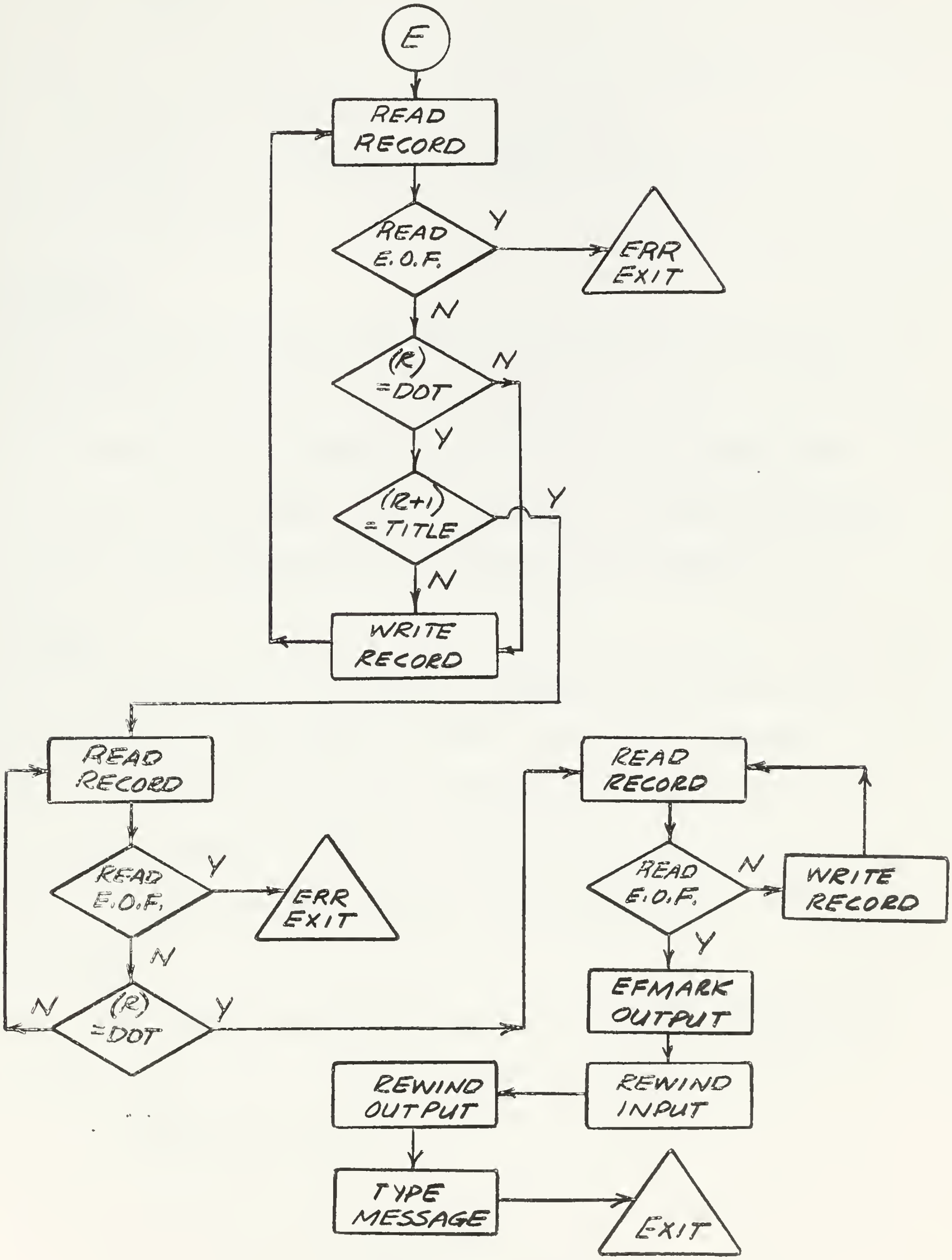


FIGURE VIII - 4

APPENDIX VIII - C

1. Identity

Title: THESIS

Category: Q

Programmers: G. H. Leach and A. J. Perrella

Organization: U. S. Naval Postgraduate School

Date: March 1964

2. Purpose

This typewriter callable service routine enables the user to print BCD records of 120 character length in a format suitable for publication in standard size, with printing on the major dimension of the page. The routine prints 47 lines, four blank lines, and a title line with incrementing page number.

3. Usage

3.1 The material to be printed in THESIS format may be on a master tape in the format of FILEIN. The routine searches for the completion tag used in FILEIN and automatically completes when this tag is found. Tapes not in FILEIN format will complete automatically when an END OF FILE mark is encountered.

3.2 Call THESIS from library tape unit 1. On the console typewriter type:

THESIS, A, B, C, D.

A is the input tape unit number

B is the output tape unit number

C is the title card tape unit number

D is the mode (0 for card images, 1 for computer written source listing)

3.3 On automatic completion, the output tape is ready to print when the routine exits. Each page will be ejected by the line printer automatically. The number of lines printed is compatible with multilith masters for this size page printed parallel to the major dimension.

3.4 The title card is read from the designated tape unit of argument 3. The format for the card may be any desired by the user with only one constraint. Spaces 25 to 32 should be left blank. These spaces will have the page number inserted, left justified. Page numbers from 1 to 999 (as necessary) will be inserted by the program.

APPENDIX VIII - D

1. Identification

Title: READ

WRITE

Category: Tape Handling

Organization: U. S. Naval Postgraduate School

Date: January 1964

2. Purpose

These program callable routines provide the Fortran programmer with an effective tool for data processing problems where large arrays must be manipulated. The routines permit writing and reading of any length records, either in binary or BCD format, directly from the desired cells which permits greater speed and flexibility than the standard Write Tape and Read Tape statements of Fortran.

3. Usage

3.1 Calling Sequence.

```
CALL READ (N, LI, LL, MODE)   CALL WRITE(N, LI, LL, MODE)
     where N is the tape unit number
           LI is the initial address (octal)
           LL is the terminal address +1 (octal)
           MODE is format control, 1 for binary, 2 for BCD
```

3.2 These routines are substantially identical to those employed by the Fortran control system, except for modified argument usage. Full parity, length, end of file, and end of

tape checking is employed.

3.3 The routine provides messages to the console typewriter in the case of tape processing errors. These are followed by sensing the typewriter for operator response in exactly the same manner as normal Fortran usage.

APPENDIX VIII - E

1. Identification

Title: WRITEMT

READMT

Category: Q (Service)

Organization: U. S. Naval Postgraduate School

Date: January 1964

2. Purpose

These routines provide the symbolic machine language programmer with the capability of writing and reading magnetic tapes of any desired record length in either binary or BCD format. The routines provide faster operation, easier processing, and greater flexibility than the use of the resident control reading and writing routines.

3. Usage

3.1 Argument handling with these routines is designed for the symbolic machine language programmer using MACHINE preamble. The arguments are entered directly rather than indirectly as with program callable routines. The argument entry and calling sequence is similar to that of SCRAP for those familiar with that assembly routine.

3.2 Reference the routine in a standard LIB statement as follows:

LIB (READMT = desired name, WRITEMT = desired name)

3.3 Calling Sequence.

Assume that the program has been tagged as RD in the LIB statement. Set up use as follows:

```
LDA (NT)          STA (RD + 1)    .
ENA (LI)          STA (RD + 2)    .
ENA (LL)          STA (RD + 3)    .
ENA (1 or 2)      STA (RD + 4)    .
SLJ4 (RD)                         .
```

where NT is the tape unit number
LI is the initial address to be read
LL is the terminal address 1
and final argument is 1 for binary or 2 for BCD tape format.

3.4 Arguments are not destroyed internally in the routines, and they need be modified only as necessary when used subsequently in the main program.

APPENDIX VIII - F

1. Identification

Title: RDMTMOD

Category: Q (Service)

Organization: U. S. Naval Postgraduate School

Date: January 1964

2. Purpose

This routine provides the symbolic machine language programmer with the capability of reading magnetic tapes of any length record in either binary or BCD format. The routine is used in conjunction with service routine WRITE. It is identical to READMT except that it provides a useful signal to the main program should an end of file be read. It may thus be used to automatically inform the main program when an end of file is read without outputting a message to the console typewriter. This feature can be of great value in many data processing problems.

3. Usage

3.1 Reference the routine with a standard LIB statement as follows:

LIB (RDMTMOD = desired name)

3.2 Use the following constant in the main program:

CON (RI = 6545646671436500B)

This is BCD representation for ENDFILE, which is used as a

signal from the tape reading routine.

3.3 Calling Sequence.

Assume that the routine has been tagged as RD in the LIB statement.

Set up the routine as follows:

```
LDA(NT)            STA(RD +1)     .
ENA(LI)            STA(RD +2)     .
ENA(LL)            STA(RD +3)     .
ENA(1 or 2)        STA(RD +4)     .
SLJ4(RD)                          .
```

where NT is the tape unit number

LI is the initial address of the read in buffer

LL is the terminal address 1

and the final argument is either 1 for binary or 2 for BCD format

3.4 After the return jump shown above, if an end of file mark has been encountered, the return to the main program will be made with the A register loaded with the constant R1. Use of the following instruction sequence is then made to sense the presence of the end of file:

```
SLJ4 (RD)                         .
SCM  (R1)     AJP(    )           .
```

where the address in the A JUMP instruction is that desired for processing necessary in the main program when an end of file is encountered.

APPENDIX IX

1. Flexowriter Tape Input.

1.1 Discussion.

1.1.1 One of the disadvantages of the satellite system is the physical distance, 5 floors, separating the satellite station from the 1604 computer. For most operations, this is only a minor inconvenience; however, the lack of magnetic tape units at the satellite station constrains a user to mount an input tape on the 1604 tape bank in order to effectively utilize the power of the system.

1.1.2 An alternate solution, using an input program on flexowriter tape to be entered at the satellite station, has been provided to alleviate this problem. Some programs will not be amenable to this sort of method due to length or because they are of a one-time nature. However, for short, general programs that only require a small amount of data to be entered - probably through CHANGE - this feature should enable a user to derive more flexibility and more efficient use of the system.

2. Operation

2.1 In order to utlize this feature, a flexowriter tape must be prepared. See the section on preparation of input tape. Although the satellite system programming is general enough to accept and transfer a flex input tape to the 1604 for a variety of uses, that is copy to magnetic tape, send control statements or perform any

other normal typewriter input function, it should be of most use as a program input medium for the MONITOR routine.

2.2 Once satellite control is established at the satellite station, position the flex tape in the 160 reader. Type whatever control statement is desired, specifying the typewriter as the input medium. For example, MONITOR, T,6. The FLEX button on Keyboard 2 of the DD 65 may now be pressed. This will cause the flex tape to be read as an input to the 1604. The FLEX and OUTPUT buttons on the DD 65 will remain lit until the input operation is complete.

> CAUTION: Until these lights go out, do not use Keyboard 1 or any of the function buttons on Keyboard 2, with the exception of LINE PRINTER, SLOW, STOP, MASTER SIGNAL, and COMM FLAG buttons. These buttons retain their normal functions.

2.3 When a STOP code is encountered on the flex input tape, the lights are turned out, all program flags reset, and normal keyboard operation is possible again.

3. Method

3.1 This routine converts an input flex tape into an 80 character BCD card image compatible with all 1604 routines using the typewriter as an input medium. Tabs on the input tape are set for spaces 7, 24, and 41. Tabs are not required; however, their use saves space on the flex tape. Delete codes and blank leader are ignored. The end of a line is signalled with a carriage

return or a STOP code. The STOP code additionally signals the 160 to exit from the FLEX routine and revert to normal operation.

3.2 When the FLEX button is pressed, the 1604 is interrupted, and a control flag is set signifying that the 160 has a BCD input for the 1604. The 160 then sets the address of the FLEX routine in SAMPLE (See the 160 executive routine coding), and loops until the 1604 is ready for the input. When signalled by the 1604, the 160 then inputs flex tape until encountering a carriage return or STOP code. It converts the flex characters to BCD and transmits an 80 character BCD card image to the 1604. When this is done, the 160 again interrupts the 1604 to set the control flag and returns to the wait loop. This sequence continues until the STOP code is found, at which time the 160 transmits the last record, resets all internal flags, and exits the FLEX routine.

3.3 During the automatic operation of the flex tape input and transfer, the system retains its sensitivity to output from the 1604, both for BCD and graph display on the DD 65 and for 1604 coded tasks.

4. Flex Tape Format.

4.1 The input flex tape format is almost completely up to the programmer; however, at no time may more than 80 characters per line be used. This will result in an error halt at the 160.

Should this halt occur, a restart at 0000 will cause another halt; however, running from the stop will cause the 160 to transmit the present contents of 80 characters to the 1604. This does not guarantee that any extra characters on that line will be ignored.

4.2 For a MONITOR input tape, the format is the same as cards. The flex tape does not have to start with a carriage return, but it is desirable to do so. The first line of the program must begin with ..JOB in the first 5 spaces. The program must be ended with two END statements on separate lines. In order to correctly finish the tape, two cases must be considered. In the first, the program input is considered to be complete - that is, either no data is required to be read by the program, or all necessary data is on the flex tape after the two END statements. In this case, the flex tape may be finished with two carriage returns and ..END followed by a STOP code punch. The second possibility is that the program may require that data be entered via the typewriter through CHANGE or some other routine. In this case, the STOP code should come immediately following the second END statement. Then, in order to exit from the MONITOR routine, type ..END and output it to the 1604 when the program is complete.

4.3 Since the flexowriter does not have an asterisk (*) on its keyboard, the 160 is programmed to accept the apostrophe ('), that is the upper case slash (/) for this symbol.

APPENDIX IX

4.4 In order to illustrate the above, the following is a demonstration program used to test the system. In this example, the coefficients of a polynomial are typed in through CHANGE and the roots are returned.

```
..job''perrelle, demonstration
      program demo
c this program is an illustration of
c what may be done using the satellite
c control system. the flexowriter tape
c contains a simple program calling one
c of the satellite system subroutines,
c change.
      dimension a(100),b(100),c(100),d(100)
1     print 100
      call change(n,a,b,epsilon)
      if(n-2) 5,5,10
5     print 50
      go to 1
10    num=n+1
      print 150,n,epsilon
      print 200,(a(i),b(i), i=1,num)
      call polyrt(a,b,n,c,d,epsilon)
      print 300
      print 400,(c(i),d(i), i=1,n)
50    format(31hequation order in error,reenter  )
100   format(27hvariables are n,a,b,epsilon    )
150   format(5x,3hn= i5,5x,9hepsilon=  f10.8,
     1 / 33hinput coefficients real       imag      )
200   format(19x,2f10.4)
300   format(16hpolynomial roots  ,/21hreal part   imag part)
400   format(2f12.6)
      end
      end
```

APPENDIX X - A

1. Identification

Title: ANALOG

Category: Hybrid Computing Aid

Programmers: A. J. Perrella and G. H. Leach

Organization: U. S. Naval Postgraduate School

Date: April 1964

2. Purpose

ANALOG is designed to provide the systems programmer with a link to the digital to analog and analog to digital equipment of the Digital Control Laboratory (Satellite Station 2). It utilizes the FORTSHARE control system to perform all necessary data transfer. The conversion equipment of the Digital Control Laboratory provides one channel A/D and two channels D/A.

3. Usage

3.1 The calling sequence is:

CALL ANALOG (NUMBER, MODE)

3.2 NUMBER is the name of the fixed point variable which is to be output to the D/A equipment, or the name of the fixed point variable, which is to be input from the A/D equipment. The range of NUMBER will be - 2047 to +2047.

3.3 MODE determines whether the operation is A/D or D/A. If MODE is 0, then the A/D equipment is interrogated for a

value to be stored in NUMBER. If MODE is 1 or 2, the value in NUMBER is sent to D/A channel 1 or 2. Any other values of MODE result in an immediate exit from the routine, with MODE set to -1.

4. Method

4.1 ANALOG examines the MODE of the calling sequence and builds a code word to be sent to the 160 satellite computer. If COMM FLAG 2 of the satellite system is not set, the routine exits, carrying a -1 in MODE; otherwise, COMM FLAG 1 is set and the code word is sent to the satellite computer, which, in turn, interprets the function. The 160 computer then performs the indicated operation and, if necessary, returns a value to the 1604. The routine then resets COMM FLAG 1.

4.2 Since the present equipment is limited to 11 bits of analog information, the range of variable permitted is -2047 to +2047. If values beyond this range are used, only the least significant 11 bits will be transferred.

4.3 Although this subroutine has been designed to work in conjunction with the satellite system, programs using it will not hang up should the satellite system be inoperative. This situation may be program sensed by examing MODE after exiting the routine. If MODE is negative, no transfer operations were attempted. MODE must then be program reset to 0, 1, or 2.

ANALOG

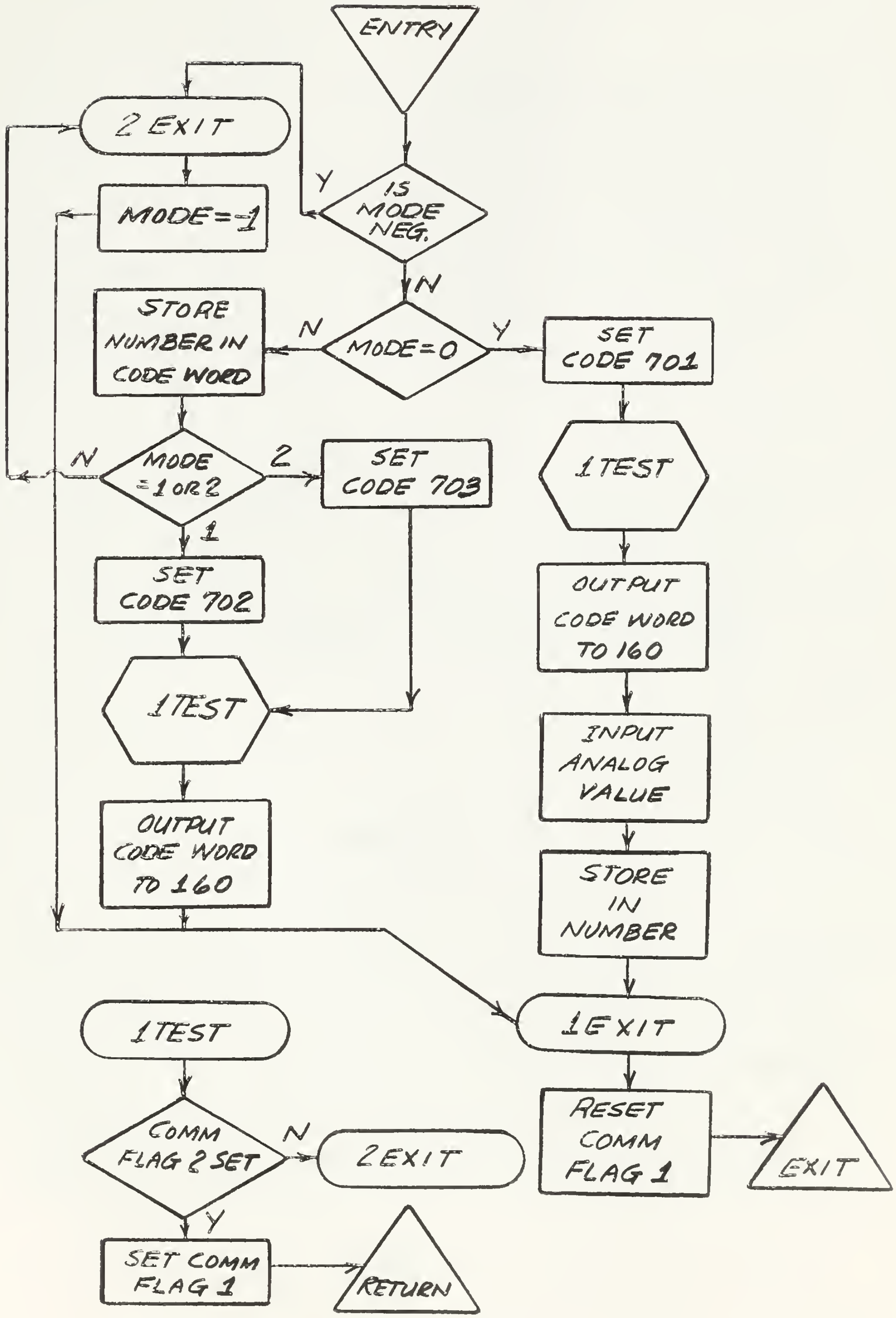


FIGURE X - 1

APPENDIX XI

1. FORTSHARE Systems Programming Additions

1.1 Main Computer Resident.

1.1.1 Task Additions.

It may be desirable to add tasks under the task processing scheme. These tasks are signalled by an interrupt from the satellite computer, followed by a 48 bit code word delivered to the main computer when requested. This code word is used to index on a directory list of tasks to select the task desired. Code word format is shown in Figure XI-1.

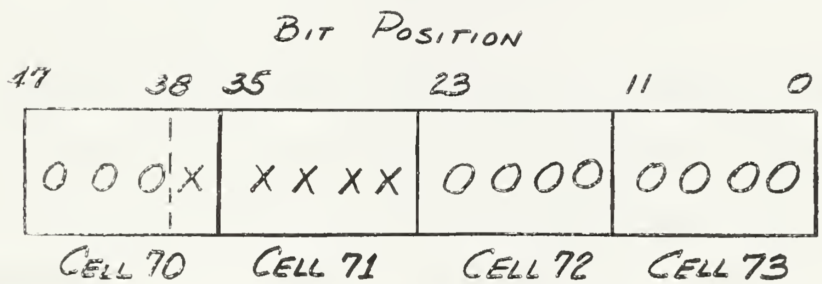


FIGURE XI-1

Code word construction for task signal.

The 48 bit pattern is assembled in 160 cells 70-73. The number in bits 24 - 38 is the value used for indexing on the task directory list. The other bits may be set at the programmers discretion in order to transmit additional information as input for the task. All tasks as presently conceived are

short and completely processed under interrupt lockout. This feature should be kept in mind when adding additional tasks. Extension of the executive control routine to provide task processing without interrupt lockout should be relatively simple, and this implementation will depend on the usage. Any closed subroutine may be added as a task. Merely add the subroutine and a directory card. The present resident bias level is set at 5000B and should be extended as necessary.

1.2 Satellite Control System.

1.2.1 Main Computer Task Coupling.

The existing control system programming for Satellite Station 2 is modular in form and may be easily extended as desired. Full decoding of Keyboard 2 of the DD-65 Display Unit has been provided in the programming, and 12 keys are, at present, unused. Coupling any of these keys to a main computer task may be accomplished by substituting the address of the program segment, which will perform the necessary processing in the cell where the specific key has been decoded. The code word may be built directly by loading cell 71 with the task number, and setting cells 70, 72, and 73 as desired, with any additional information required by the task. The program segment should then jump to location INT in Keyboard 2, Part 1. The program will then return to the executive control portion of the system programming, and

the satellite will immediately be sensitive to input from the main computer. Sufficient examples are provided in the coding (19) to demonstrate varied use of this procedure.

1.2.2 Satellite Tasks.

Intercomputer communications have been enhanced by also providing the capability of task direction from the main computer to the satellite computer. This may be accomplished at any time by setting COMM FLAG 1 for the appropriate Satellite Station (channels 5-6 for Satellite Station 2), and delivering a 48 bit code word to the satellite computer. The code word will consist of the number 07XXB in the lower 12 bits, and the upper 36 bits may carry additional information as desired. At present, only five such tasks have been utilized and decoded, but a linkage has been provided for further decoding of up to 63 tasks. This link is low core cell 47. The address of the program segment performing further decoding may be placed in this cell. Examination of location DECODE in the satellite control coding (19) demonstrates this feature.

1.2.3 Additional Satellite Programming Feature.

In order to provide a linkage in the 160 for a transient or periodic task, a low core cell (SAMPLE) has been provided. This cell normally contains the address of EXEC, but may be changed to the address of any routine desired; and that routine will be

executed each time the satellite enters its wait loop during periods of inactivity. Such tasks may be sensing the DD-65 for radar targets, sampling the A/D converters, or sensing the main computer ready for an input from the satellite. The task should exit to EXEC when completed in order to maintain system sensitivity. An example of the use of this link may be found in location FLEX in the satellite control coding (19), where the address in SAMPLE is modified by programming.